Garuda User's Handbook

First Edition

GARUDA User's Handbook

Version 1.0

19 May 2011

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

Acknowledgement

Quite a few people are responsible, directly and indirectly, in the successful completion of this book to whom we owe our sincere gratitude. First of all, we would like to thank our respected Executive Director, Dr. Sarat Chandra Babu, for always being so enthusiastic and encouraging towards us. Special thanks to Dr. Subrata Chattopadhyay**,** for his direction and guidance in all our Garuda activities.

Thanks are also due to Mr. R. Sridharan for having the faith in us and increasing our confidence time and again. We also wish to thank Dr. Prahalad Rao and Ms. Mangala, for lending their support to us whenever required.

Many thanks to Ms. Divya and the whole Garuda operational team for their unending efforts for the Garuda project and in turn helping us in all our activities.

Finally, we wish to express our appreciation for Ms. Prachi for helping us to construct meaningful grammatical sentences, which hopefully are also error-free.

Authors:

S. Henry Sukumar & Santhosh J

GARUDA Operational Team.

Table of Contents

Introduction

GARUDA is India's National Grid Computing initiative bringing together academic, scientific and Research communities for developing their data and compute intensive applications with guaranteed QoS. GARUDA grid is an aggregation of resources comprising of computational nodes, mass storage and scientific instruments distributed across the country.

The nation-wide computational grid “GARUDA” spans across 17 cities with an aim to bring Distributed/Grid networked infrastructure to academic labs, research labs and industries in India.

Reference: ***http://garudaindia.in***

Who This Book Is For

This book is intended for the people who are interested in using the GARUDA Grid for meeting their computational needs. It expects basic knowledge about grid computing and UNIX. It provides an easy to follow user’s guide of Garuda with detailed step-by-step illustration of getting access to the Garuda and using it.

Chapter Organization

The **first** chapter focuses on steps to access Garuda i.e. getting a grid certificate and a Garuda account.

The **second** and **third** chapters explain about the access mechanisms available in Garuda i.e. the Grid Job Submission Portal and the Command Line Interface.

The **fourth** chapter explains about the usage of the commonly used commands for the grid services. It is divided into three sections, which deal with Credential Management, Data Management and Job Management.

The **fifth** chapter explains about the usage of Garuda Grid FTP client software. Garuda GridFTP Client is a GUI for performing file transfers using the GridFTP protocol.

The **sixth** chapter explains about the usage of commonly used commands of Garuda Storage Resource Manager.

The chapter **seven** explains about the mechanism by which users can report any problem they face while using Garuda.

Chapter 1

Steps to access GARUDA

Chapter 1: Steps to access GARUDA

Getting access to GARUDA involves two steps:

Step 1: Get Grid User Certificate from IGCA

Visit ***http://ca.garudaindia.in/index.php/certificate/request-a-new-certificate/***.

Follow the steps mentioned below to apply for a user certificate.

a. Download and fill up IGCA User Application Form : ***http://ca.garudaindia.in/docs/IGCA_User_Form_V1.pdf***

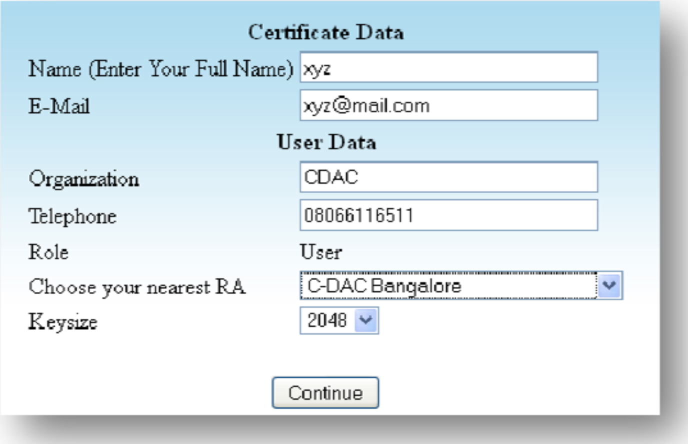
b. Register online to get a request no. : ***https://ca.garudaindia.in/cgi-bin/pub/pki?cmd=basic_csr*** (*Supported browsers: Internet Explorer and Mozilla Firefox only*). As depicted in Fig. 1.1: Enter the necessary details.

- **Name**: Full name of the applicant.
- **E-Mail**: Valid Email address for communication.
- **Organization**: Name of your institution/organization
- **Telephone**: Valid telephone no. for communication.
- **Role**: Default value "User", for user certificate.
- **Choose your nearest RA**: Choose your RA for face to face meeting.
- **Key size**: Default value "2048", is the strength of certificate.

***IGCA**: **I**ndian **G**rid **C**ertification **A**uthority*

***RA**: **R**egistration **A**uthority.*

Refer Appendix-I for glossary and terms



Certificate Data

Name (Enter Your Full Name) xyz

E-Mail xyz@mail.com

User Data

Organization CDAC

Telephone 08066116511

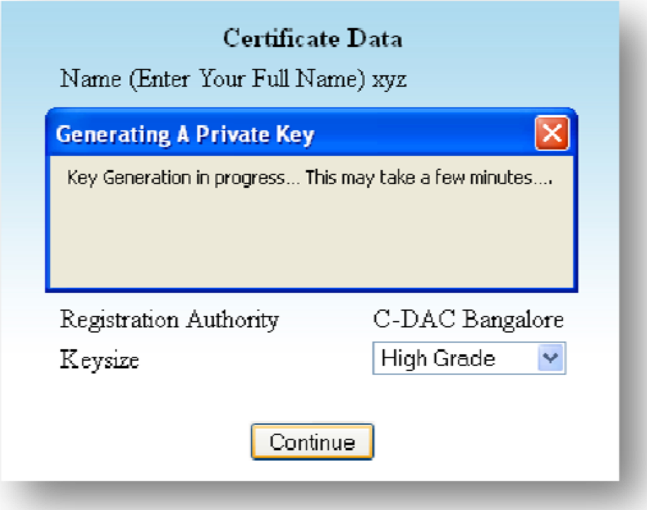
Role User

Choose your nearest RA C-DAC Bangalore

Keysize 2048

Fig. 1.1: Online Registration

c. Clicking "Continue", will generate a private key as shown in Fig. 1.2.



Certificate Data

Name (Enter Your Full Name) xyz

Generating A Private Key

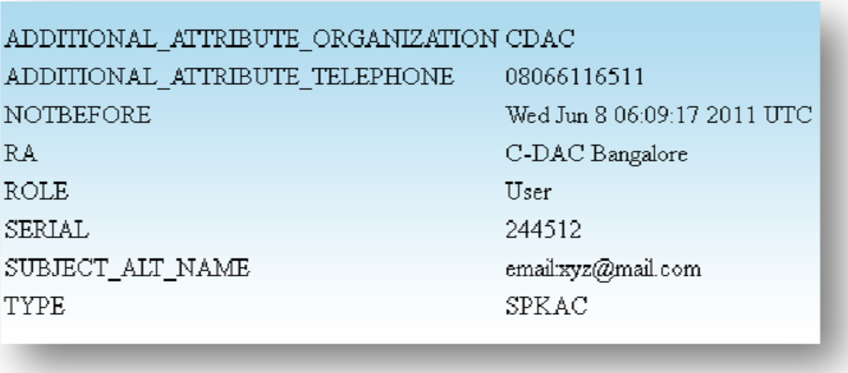
Key Generation in progress... This may take a few minutes....

Registration Authority C-DAC Bangalore

Keysize High Grade

Fig. 1.2: Key generation process

d. Fig. 1.3: Shows the details of your request. Note down the ***SERIAL No***. in your application form.



ADDITIONAL_ATTRIBUTE_ORGANIZATION	CDAC
ADDITIONAL_ATTRIBUTE_TELEPHONE	08066116511
NOTBEFORE	Wed Jun 8 06:09:17 2011 UTC
RA	C-DAC Bangalore
ROLE	User
SERIAL	244512
SUBJECT_ALT_NAME	email:xyz@mail.com
TYPE	SPKAC

Fig. 1.3: Get your request no "SERIAL"

e. Choose your nearest RA from the below link and arrange for a Face to Face meeting. (***http://ca.garudaindia.in/index.php/ra/contact-ra/***)

f. Mail the scanned copy of your application form and ID proof to igca@cdacb.ernet.in

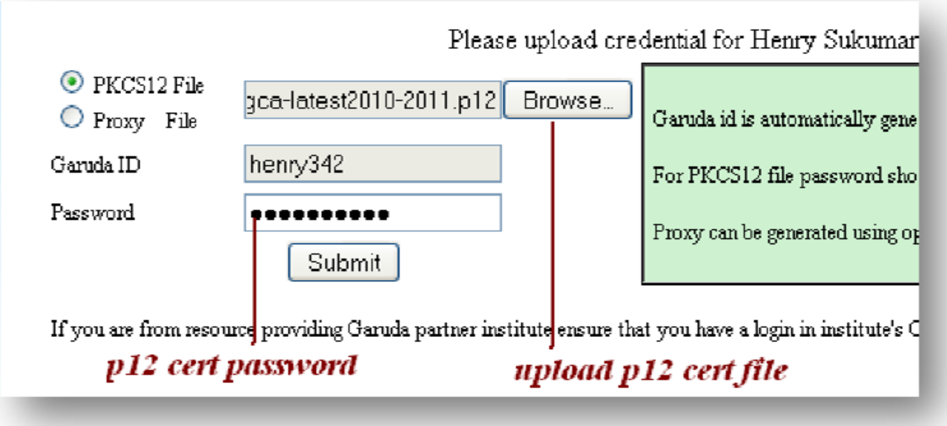
g. Your certificate information will be mailed to you.

Note: *Don't uninstall browser or format your machine until your certificate is issued.*

Refer ***http://ca.garudaindia.in*** *for more information.*

Step2: GARUDA Registration

1. Visit ***http://voms.garudaindia.in***. Make sure you have your IGCA certificate uploaded in the web browser.
2. Upload your IGCA certificate file (.p12 format) using the browse button as shown in Fig. 1.4 and enter your password (the same password which you entered while exporting your certificate from the browser). "Garuda ID" is generated automatically by the system.



Please upload credential for Henry Sukumar

◉ PKCS12 File
○ Proxy File

gca-latest2010-2011.p12

Garuda ID henry342

Password ●●●●●●●●●●●

Garuda id is automatically gene

For PKCS12 file password sho

Proxy can be generated using op

If you are from resource providing Garuda partner institute ensure that you have a login in institute's C

p12 cert password

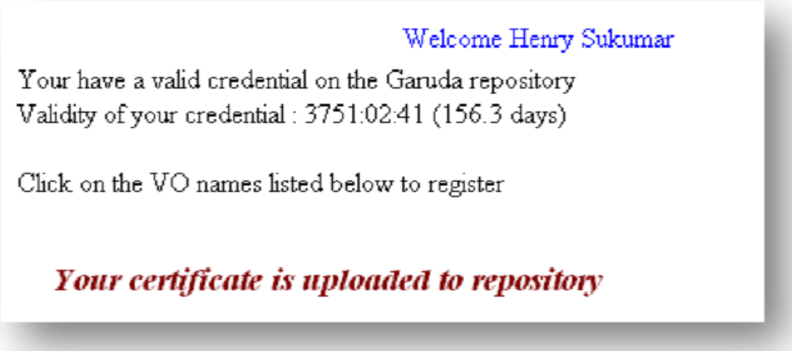
upload p12 cert file

Fig. 1.4: Garuda Credential upload.

What if I don't have a '.p12' certificate file?
***Sol**: Visit*
http://ca.garudaindia.in/index.php/information/faq/export-the-certificates-obtain-pfxps12-file/

Refer http://voms.garudaindia.in for more information.

3. The image below shows the acknowledgement page that you will receive once your certificate has been successfully uploaded to the GARUDA credential repository.



Welcome Henry Sukumar

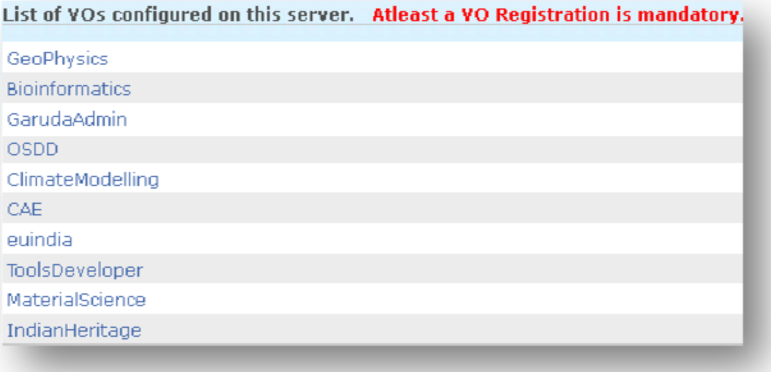
Your have a valid credential on the Garuda repository
Validity of your credential : 3751:02:41 (156.3 days)

Click on the VO names listed below to register

Your certificate is uploaded to repository

Fig. 1.5: GARUDA Credential upload acknowledgement

4. Now you can see a list of available VO's. Select your VO Name from the list. At least one VO registration is mandatory. ***E.g.: It can be the name of your working domain.***



List of VOs configured on this server. Atleast a VO Registration is mandatory.

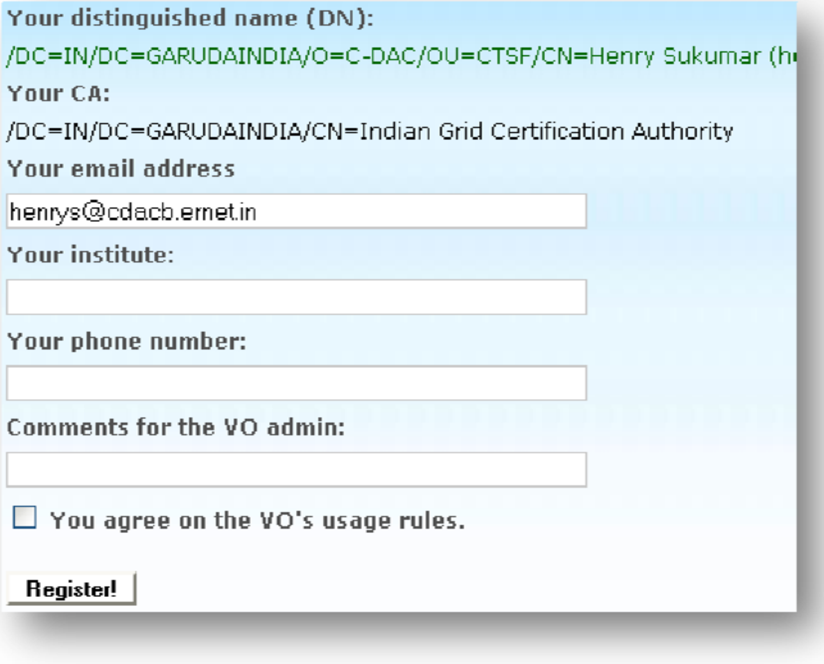
- GeoPhysics
- Bioinformatics
- GarudaAdmin
- OSDD
- ClimateModelling
- CAE
- euindia
- ToolsDeveloper
- MaterialScience
- IndianHeritage

Fig. 1.6: Select your VO

What if my VO is not listed here?

Sol: Send a mail to voms@cdacb.ernet.in

5. Fill the form as shown in Fig. 1.7 and click on "***Register***" button.



Your distinguished name (DN):

/DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (h

Your CA:

/DC=IN/DC=GARUDAINDIA/CN=Indian Grid Certification Authority

Your email address

henrys@cdacb.ernet.in

Your institute:

Your phone number:

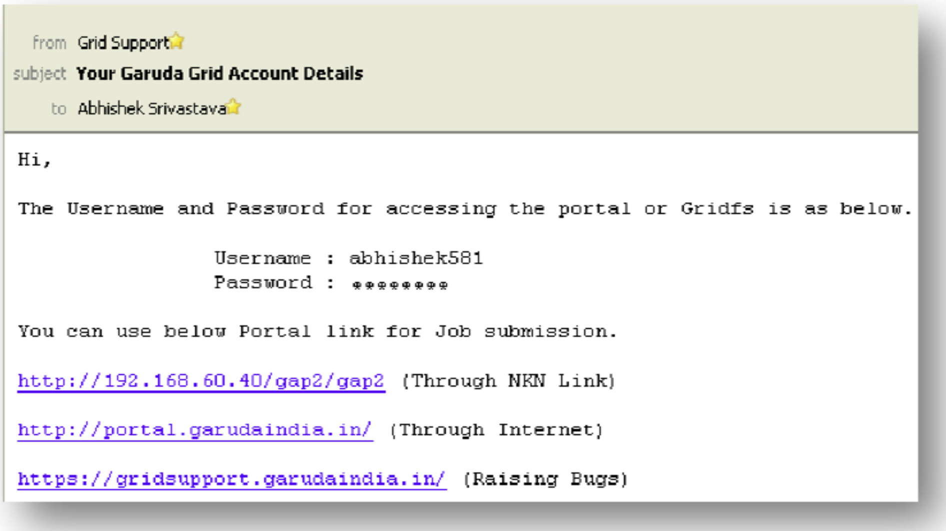
Comments for the VO admin:

□ You agree on the VO's usage rules.

Fig. 1. 7: VO Registration form

***VO** stands for **V**irtual **O**rganization*

6. As shown in Fig. 1.8, you will receive an e-mail from GARUDA Admin containing information about your access details, which is hereafter referred to, as your "***GARUDA Account***".



from Grid Support
subject **Your Garuda Grid Account Details**
to Abhishek Srivastava

```
Hi,

The Username and Password for accessing the portal or Gridfs is as below.

                Username : abhishek581
                Password : ********

You can use below Portal link for Job submission.

http://192.168.60.40/gap2/gap2 (Through NKN Link)

http://portal.garudaindia.in/ (Through Internet)

https://gridsupport.garudaindia.in/ (Raising Bugs)
```

Fig. 1.8: Mail about your GARUDA account

> ***Note:*** *Save this mail for your future reference.*
> *For Garuda account related issues:* ***grid-help@cdacb.ernet.in***

Chapter 2

Accessing GARUDA via Grid Portal

Chapter 2: Accessing GARUDA via Grid Portal

What is job submission Portal?

Grid portal is used for submitting sequential and parallel jobs to the grid. Grid is a cluster of heterogeneous workstations located at geographically distant locations, with the ultimate aim of sharing computing resources. The portal provides a web interface for viewing the resources, for submitting and monitoring jobs.

What is the URL to access Grid Portal via Internet?

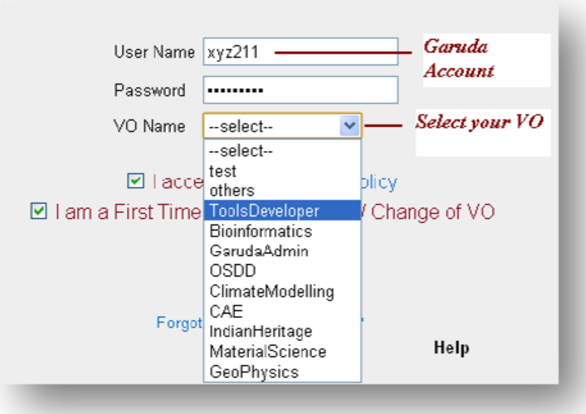
http://portal.garudaindia.in

How to access portal via NKN (National Knowledge Network)?

http://192.168.60.40/gap2

Which username and password should I use to access portal?

You need a GARUDA account for accessing Grid Portal. Refer ***Fig. 1.8 from chapter 1***.



User Name xyz211

Password •••••••••

VO Name --select--

☑ I acce… …olicy

☑ I am a First Time … Change of VO

Fig. 2.1: Shows the login form to access Grid Portal

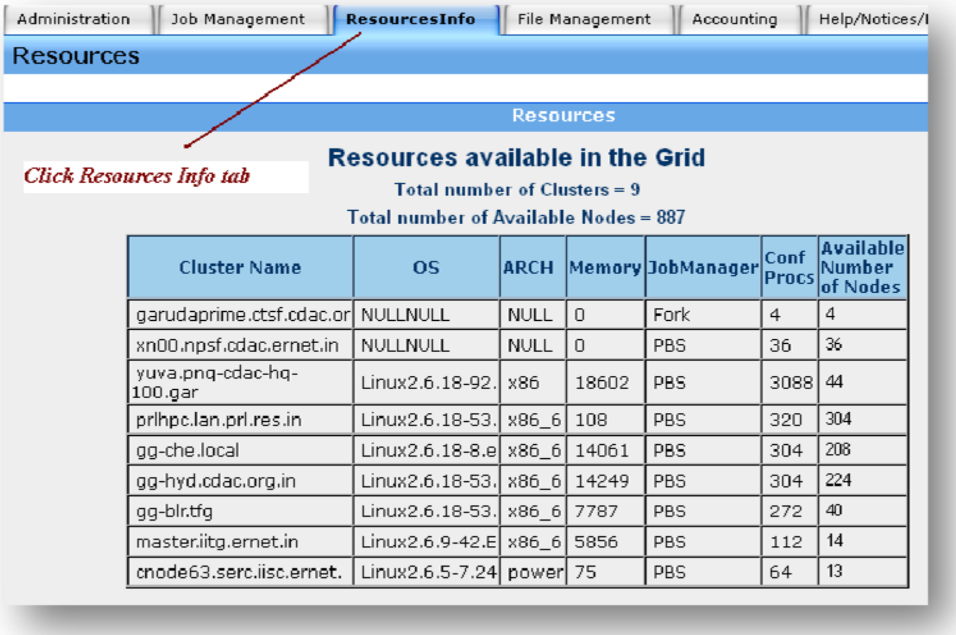
Note: *You need to select a VO Name, only for the first time.*

Which VO should I select during my login to Grid Portal?

Select the VO that you had registered with GARUDA in ***Chapter 1***.

How to view the list of available resources in GARUDA?

Click "***Resource Info***" tab as shown in Fig. 2.2. to view the available resources in the Grid.



Resources available in the Grid

Total number of Clusters = 9

Total number of Available Nodes = 887

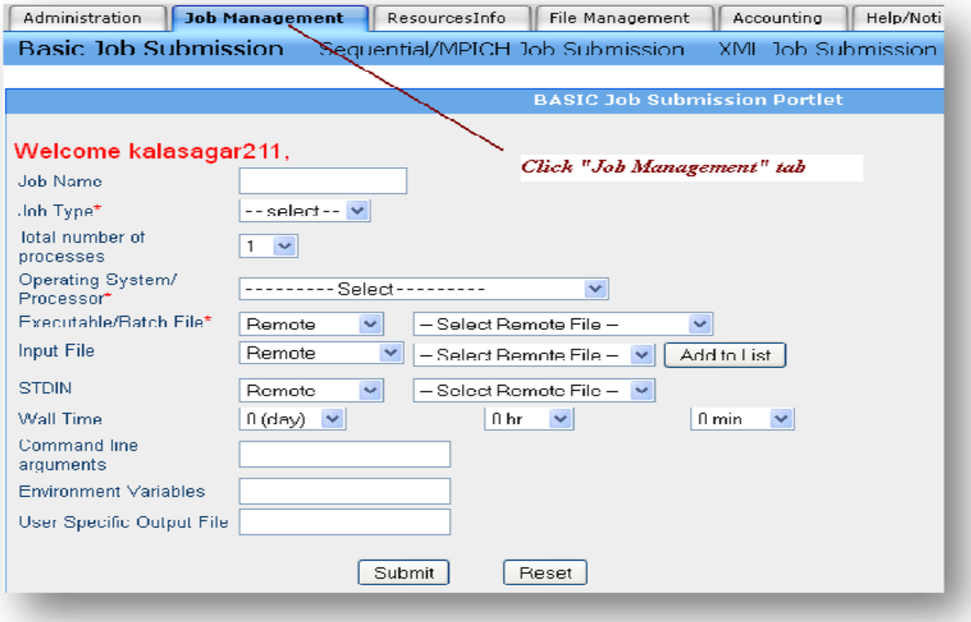
Cluster Name	OS	ARCH	Memory	JobManager	Conf Procs	Available Number of Nodes
garudaprime.ctsf.cdac.or	NULLNULL	NULL	0	Fork	4	4
xn00.npsf.cdac.ernet.in	NULLNULL	NULL	0	PBS	36	36
yuva.pnq-cdac-hq-100.gar	Linux2.6.18-92.	x86	18602	PBS	3088	44
prlhpc.lan.prl.res.in	Linux2.6.18-53.	x86_6	108	PBS	320	304
gg-che.local	Linux2.6.18-8.e	x86_6	14061	PBS	304	208
gg-hyd.cdac.org.in	Linux2.6.18-53.	x86_6	14249	PBS	304	224
gg-blr.tfg	Linux2.6.18-53.	x86_6	7787	PBS	272	40
master.iitg.ernet.in	Linux2.6.9-42.E	x86_6	5856	PBS	112	14
cnode63.serc.iisc.ernet.	Linux2.6.5-7.24	power	75	PBS	64	13

Fig. 2.2: Grid Portal: Resource Information

Grid Portal: ***http://portal.garudaindia.in***

How to submit jobs using Grid Portal?

Click "***Job Management***" tab to submit jobs as shown in Fig. 2.3.



Click "Job Management" tab

Welcome kalasagar211,

Job Name

Job Type* -- select --

Total number of processes 1

Operating System/ Processor* ---------Select---------

Executable/Batch File* Remote — Select Remote File —

Input File Remote — Select Remote File —

STDIN Remote — Select Remote File —

Wall Time 0 (day) 0 hr 0 min

Command line arguments

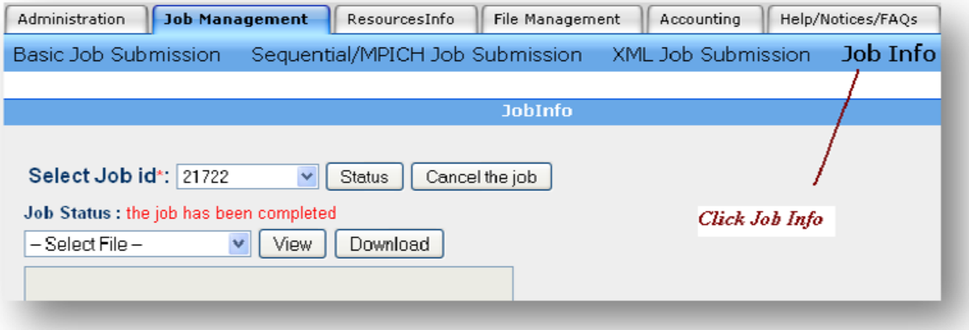
Environment Variables

User Specific Output File

Fig. 2.3: Grid Portal: Job Management Section

How to check the status of the job and view or download output/error file?

Click ***"Job Management"*** >>>> ***"Job Info"*** as shown in Fig. 2.4.



Job Status : the job has been completed

Fig. 2.4: Get Job Status and view or download output/error files

How to manage files stored in GRIDFS (GARUDA Head node) and GSRM (Garuda Storage Resource Manager) via Grid Portal?

Click "***File Management***" tab as shown in Fig. 2.5.

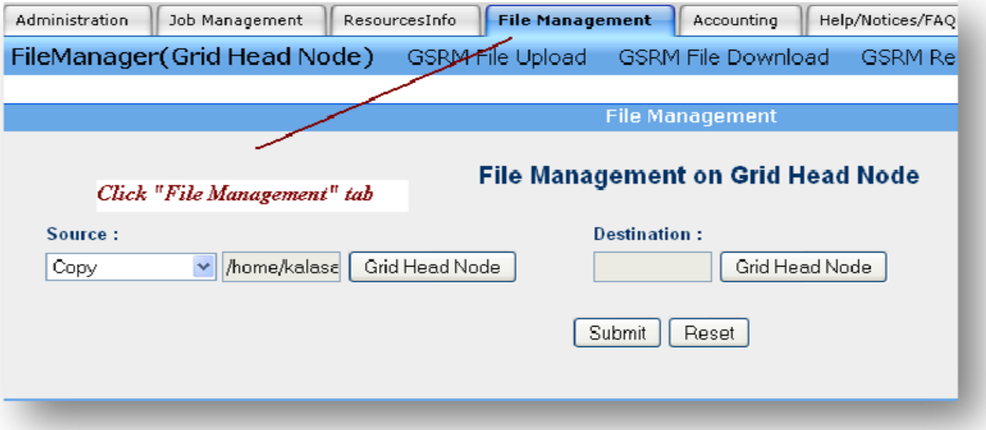


Fig. 2.5: Grid Portal: File Management Section

Chapter 3

Accessing GARUDA via Command line

Chapter 3: Accessing GARUDA via Command line

GARUDA Grid is connected across the country through NKN network of 1Gbps.

Steps to access via NKN:

1. An interface node has been installed in each partner institution, which is hereafter referred to as "***GARUDA INTERFACE NODE***". Any user, who wants to connect to GARUDA, should get an access to ***GARUDA INTERFACE NODE*** at his/her institution.

2. Contact your local system administrator to get the information about the GARUDA INTERFACE NODE located at your institution.
   Ex: IP address, reachability to ***gridfs.ctsf.cdac.org.in (GRIDFS).***

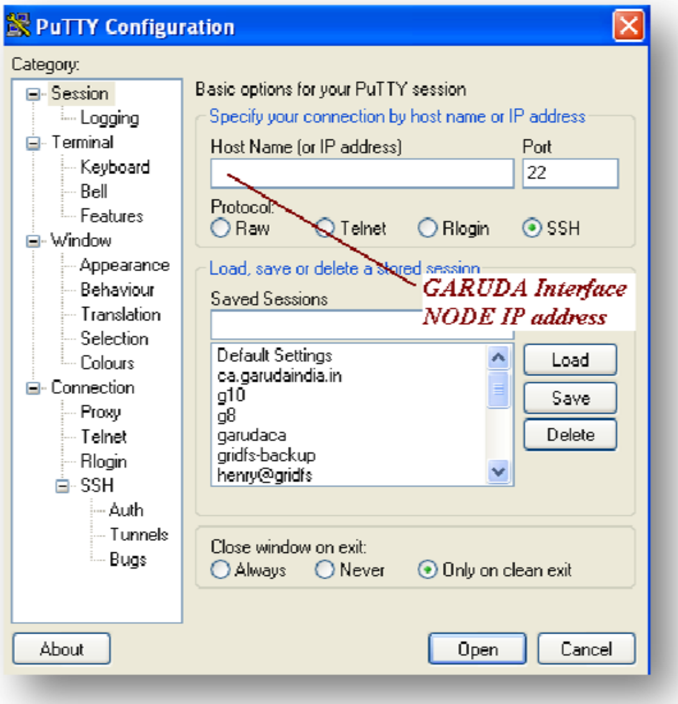
3. Get the login access to GARUDA INTERFACE NODE in your institution;

4. Make sure you have IGCA certificate and GARUDA Grid account. (Refer Chapter 1)

For WINDOWS Users:

A. Download the program called "***puTTY***" from the below link:
   http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

B. Double click and open ***putty.exe***. This will open the puTTY configuration as shown in Fig. 3.1.



Host Name (or IP address)

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

Saved Sessions

Default Settings
ca.garudaindia.in
g10
g8
garudaca
gridfs-backup
henry@gridfs

Close window on exit: ○ Always ○ Never ● Only on clean exit

GARUDA Interface NODE IP address

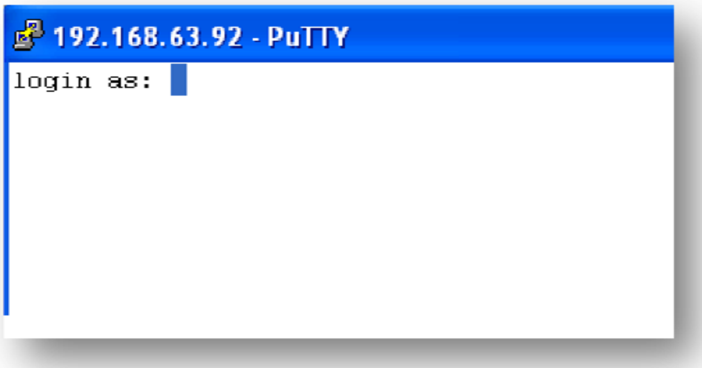
Fig. 3.1: PUTTY Interface for SSH connection

C. Enter the “***GARUDA INTERFACE NODE IP ADDRESS”*** in the “***Hostname [or IP address]***” text-box. Refer Fig. 3.1.

D. Protocol: ***Select SSH***. (Refer above Fig. 3.1)

E. Click “***open***” button, you can see a terminal as shown in Fig. 3.2

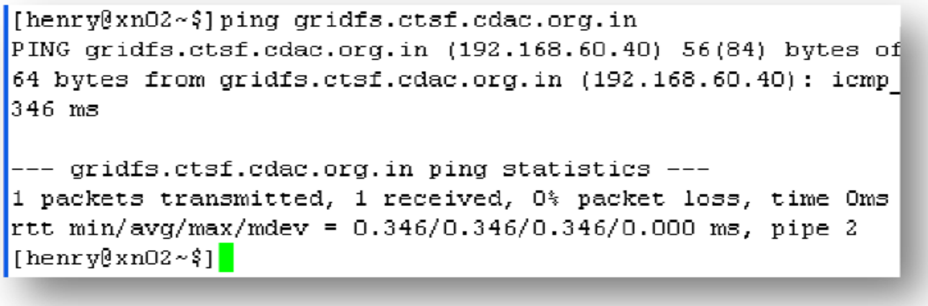
F. Enter the username and password provided by the system administrator in your institute/organization.



```
login as: 
```

Fig. 3.2: Login to GARUDA Interface Node.

G. Make sure that you can reach GRIDFS using ***PING*** command as shown in Fig. 3.3. (***gridfs.ctsf.cdac.org.in*** – GARUDA Grid Head Node).



```
[henry@xn02~$]ping gridfs.ctsf.cdac.org.in
PING gridfs.ctsf.cdac.org.in (192.168.60.40) 56(84) bytes of
64 bytes from gridfs.ctsf.cdac.org.in (192.168.60.40): icmp_
346 ms

--- gridfs.ctsf.cdac.org.in ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.346/0.346/0.346/0.000 ms, pipe 2
[henry@xn02~$]
```

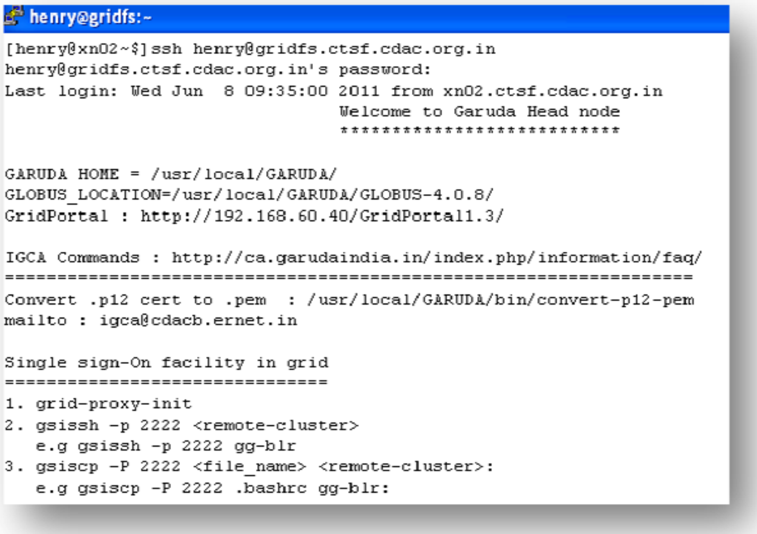
Fig. 3.3: Check for reachability to gridfs

What is PuTTY?

PuTTY is a program that allows windows users to connect to UNIX like machines.

H. Use ***SSH*** command to login to the ***GARUDA HEAD NODE (gridfs.ctsf.cdac.org.in)*** with your username and password (Refer Chapter 1: Step 2) obtained from the GARUDA.
The command is given below. Refer Fig. 3.4.

SSH <username>@gridfs.ctsf.cdac.org.in



```
[henry@xn02~$]ssh henry@gridfs.ctsf.cdac.org.in
henry@gridfs.ctsf.cdac.org.in's password:
Last login: Wed Jun  8 09:35:00 2011 from xn02.ctsf.cdac.org.in
                                Welcome to Garuda Head node
                                ***************************

GARUDA HOME = /usr/local/GARUDA/
GLOBUS_LOCATION=/usr/local/GARUDA/GLOBUS-4.0.8/
GridPortal : http://192.168.60.40/GridPortal1.3/

IGCA Commands : http://ca.garudaindia.in/index.php/information/faq/
==================================================================
Convert .p12 cert to .pem  : /usr/local/GARUDA/bin/convert-p12-pem
mailto : igca@cdacb.ernet.in

Single sign-On facility in grid
===============================
1. grid-proxy-init
2. gsissh -p 2222 <remote-cluster>
   e.g gsissh -p 2222 gg-blr
3. gsiscp -P 2222 <file_name> <remote-cluster>:
   e.g gsiscp -P 2222 .bashrc gg-blr:
```

Fig. 3.4: Login to Gridfs machine

GARUDA HEAD NODE: gridfs.ctsf.cdac.org.in (192.168.60.40)

- *This node is called as **GRIDFS**.*
- *You need a GARUDA Account to access this node. (Chapter 1)*
- *This node is reachable only via NKN Network*
- *As per the policy, internet users cannot login to GRIDFS. They have to use Grid Portal. (Chapter 2)*

Chapter 4

Grid
Command reference

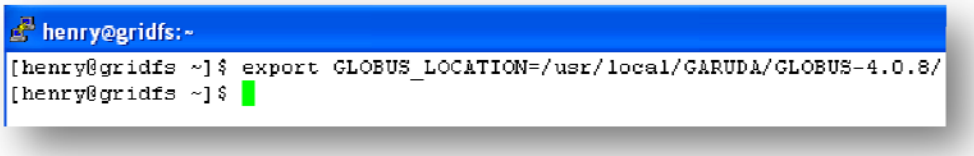
Chapter 4: Grid Command reference

What is Globus Toolkit?

Globus Toolkit is a popular grid middleware among the scientific grid computing platforms. Globus incorporates not just **job management** but also **storage management, security provisioning, data movement, monitoring.** It is also a toolkit for developing additional services based on the same infrastructure, including agreement negotiation, notification mechanisms; trigger services, and information aggregation.

How to set necessary environment variables in command line?

Fig. 4.1 depicts the necessary command to set up Globus environment variable (GLOBUS_LOCATION) at gridfs machine. Note: The installation path of globus may vary on different clusters and environment variable would have been **already set by the administrator**. If not set, users can set as shown in the below fig. 4.1

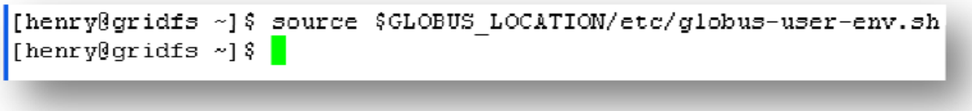


```
[henry@gridfs ~]$ export GLOBUS_LOCATION=/usr/local/GARUDA/GLOBUS-4.0.8/
[henry@gridfs ~]$
```

Fig. 4.1: Setting globus environment variable

Exporting Globus commands:

As shown in Fig. 4.2, ***source*** command is used for exporting the globus commands and libraries in "PATH" environment variable in gridfs machine. Note: The installation path of globus may vary on different clusters.

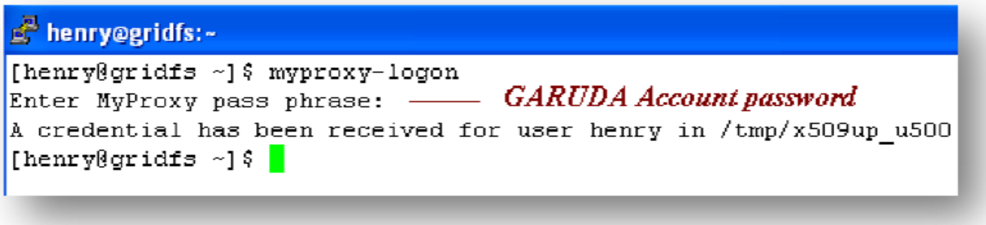


```
[henry@gridfs ~]$ source $GLOBUS_LOCATION/etc/globus-user-env.sh
[henry@gridfs ~]$
```

Fig. 4.2: Exporting globus commands and libraries in path

How to download my credentials for using grid?

myproxy-logon command is used to download user credentials from the credentials repository as shown in Fig. 4.3. Later these credentials can be used for accessing grid services.



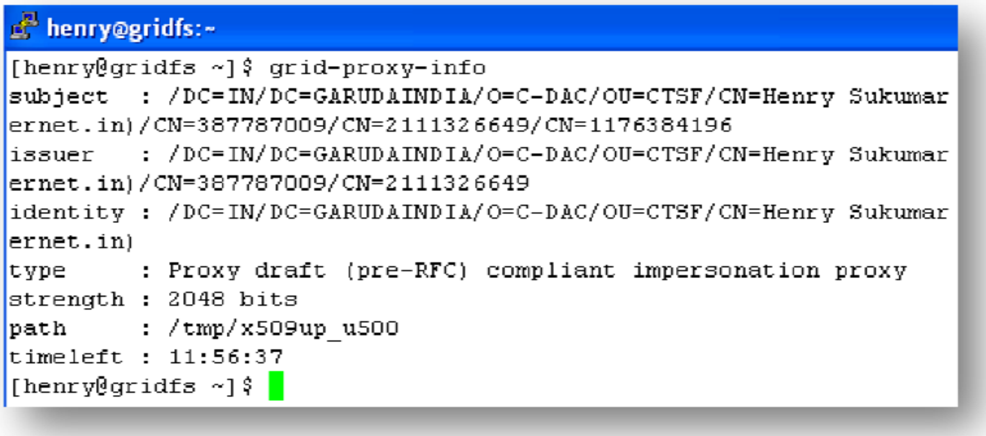
```
[henry@gridfs ~]$ myproxy-logon
Enter MyProxy pass phrase:
A credential has been received for user henry in /tmp/x509up_u500
[henry@gridfs ~]$
```

GARUDA Account password

Fig. 4.3: Download your proxy for grid usage

How to check the status of my credential/proxy?

grid-proxy-info command is used to verify the status of user credentials as shown in Fig. 4.4

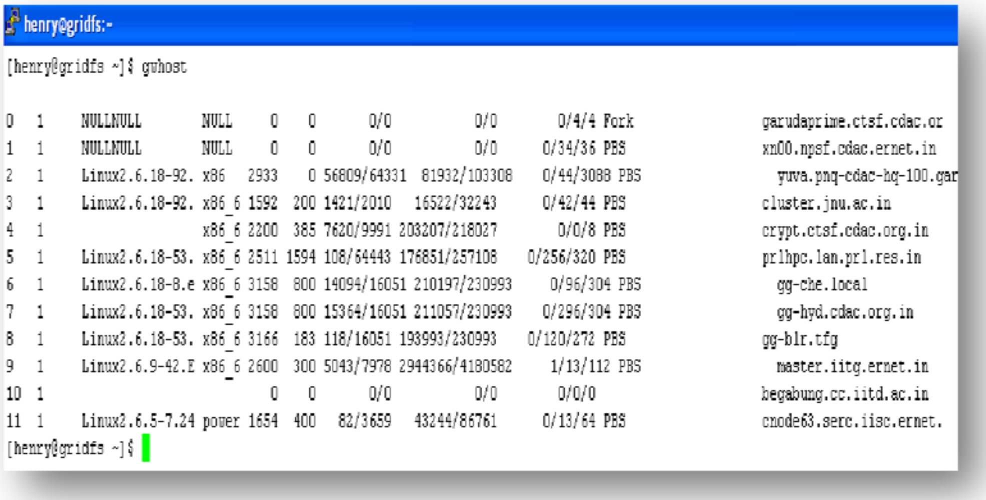


```
[henry@gridfs ~]$ grid-proxy-info
subject  : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar
ernet.in)/CN=387787009/CN=2111326649/CN=1176384196
issuer   : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar
ernet.in)/CN=387787009/CN=2111326649
identity : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar
ernet.in)
type     : Proxy draft (pre-RFC) compliant impersonation proxy
strength : 2048 bits
path     : /tmp/x509up_u500
timeleft : 11:56:37
[henry@gridfs ~]$
```

Fig. 4.4: grid-proxy-info command to check the validity of your credential

How to check the list of available resources in grid?

gwhost command is used to check the available resources in grid as shown in Fig. 4.5.

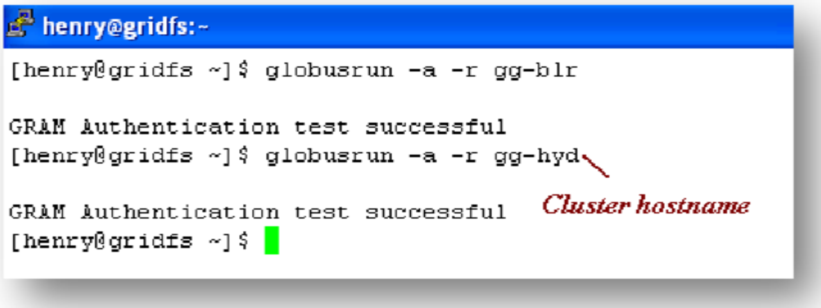


```
henry@gridfs:~
[henry@gridfs ~]$ gwhost

0  1    NULLNULL        NULL     0    0         0/0             0/0      0/4/4 Fork    garudaprime.ctsf.cdac.or
1  1    NULLNULL        NULL     0    0         0/0             0/0    0/34/36 PBS     xn00.npsf.cdac.ernet.in
2  1    Linux2.6.18-92. x86   2933    0 56809/64331    81932/103308  0/44/3088 PBS       yuva.pnq-cdac-hq-100.gar
3  1    Linux2.6.18-92. x86_6 1592  200   1421/2010     16522/32243    0/42/44 PBS     cluster.jnu.ac.in
4  1                    x86_6 2200  385   7620/9991   203207/218027      0/0/8 PBS     crypt.ctsf.cdac.org.in
5  1    Linux2.6.18-53. x86_6 2511 1594  108/64443   176851/257108  0/256/320 PBS     prlhpc.lan.prl.res.in
6  1    Linux2.6.18-8.e x86_6 3158  800 14094/16051  210197/230993   0/96/304 PBS       gg-che.local
7  1    Linux2.6.18-53. x86_6 3158  800 15364/16051  211057/230993  0/296/304 PBS       gg-hyd.cdac.org.in
8  1    Linux2.6.18-53. x86_6 3166  183  118/16051   193993/230993  0/120/272 PBS     gg-blr.tfg
9  1    Linux2.6.9-42.E x86_6 2600  300  5043/7978  2944366/4180582   1/13/112 PBS       master.iitg.ernet.in
10 1                             0    0         0/0             0/0      0/0/0         begabung.cc.iitd.ac.in
11 1    Linux2.6.5-7.24 power 1654  400    82/3659     43244/86761    0/13/64 PBS     cnode63.serc.iisc.ernet.
[henry@gridfs ~]$
```

Fig. 4.5: gwhost command to check the available resources in grid

How to check my authentication to remote cluster?

globusrun command is used to check user authentication to remote cluster as shown in Fig. 4.6.

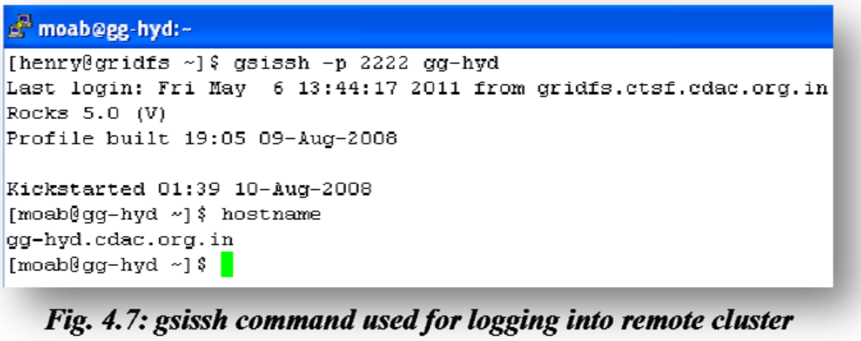


```
henry@gridfs:~
[henry@gridfs ~]$ globusrun -a -r gg-blr

GRAM Authentication test successful
[henry@gridfs ~]$ globusrun -a -r gg-hyd

GRAM Authentication test successful
[henry@gridfs ~]$
```

Fig. 4.6: globusrun command to check user authentication to remote cluster.

How will I login to remote cluster using my proxy certificate?

You can login to the remote cluster using the single sign-on feature in Garuda. **Single sign-on** (**SSO**) is a property of access control of multiple related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them. Single sign-on is achieved using the command ***gsissh*** as shown in Fig. 4.7.

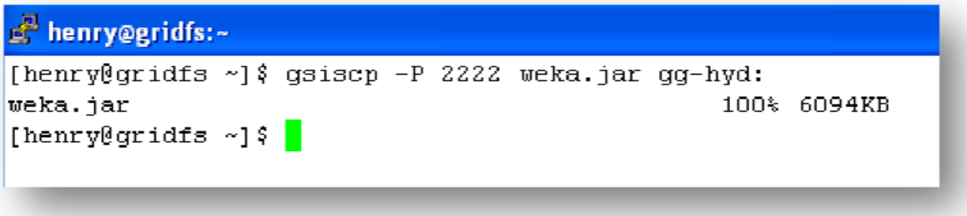


```
[henry@gridfs ~]$ gsissh -p 2222 gg-hyd
Last login: Fri May  6 13:44:17 2011 from gridfs.ctsf.cdac.org.in
Rocks 5.0 (V)
Profile built 19:05 09-Aug-2008

Kickstarted 01:39 10-Aug-2008
[moab@gg-hyd ~]$ hostname
gg-hyd.cdac.org.in
[moab@gg-hyd ~]$ 
```

Fig. 4.7: gsissh command used for logging into remote cluster

How to copy data to remote cluster using GSISCP?

gsiscp command is used to transfer data to remote cluster as shown in Fig. 4.8.



```
[henry@gridfs ~]$ gsiscp -P 2222 weka.jar gg-hyd:
weka.jar                                      100%  6094KB
[henry@gridfs ~]$ 
```

Fig. 4.8: gsiscp command is used for transferring data to remote cluster.

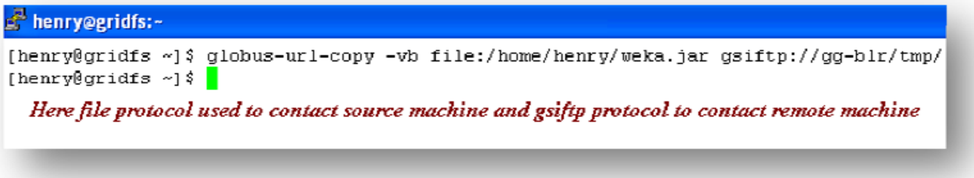
GSISSH and GSISCP:

- *Use '**-p**' (small p) option for **GSISSH***
- *Use '**-P**' (capital P) option for **GSISCP***
- *2222: GSISSH service port.*

How to copy data to remote cluster using GRIDFTP?

The ***GridFTP*** protocol provides secure, robust, fast and efficient transfer of (especially bulk) data. Data can be transferred from,

Transferring files from local machine (source) to remote machine (destination): In this case, file protocol is used to contact the source machine and gsiftp protocol should be used to contact remote machine.

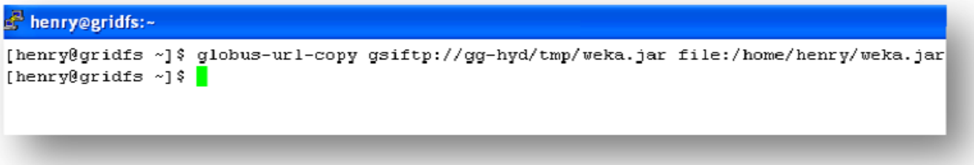


```
[henry@gridfs ~]$ globus-url-copy -vb file:/home/henry/weka.jar gsiftp://gg-blr/tmp/
[henry@gridfs ~]$
```

Here file protocol used to contact source machine and gsiftp protocol to contact remote machine

Fig. 4.9: globus-url-copy command – file to gsiftp

Transferring files from remote machine to local machine:



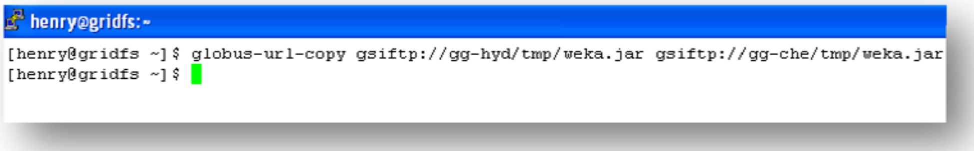
```
[henry@gridfs ~]$ globus-url-copy gsiftp://gg-hyd/tmp/weka.jar file:/home/henry/weka.jar
[henry@gridfs ~]$
```

Fig. 4.10: globus-url-copy command – gsiftp to file

*For transferring data to **GSRM** (Garuda Storage Resource Manager): Refer chapter 6.*

Third Party Transfer:

Transferring files between machine B and machine C from machine A.
E.g. transferring files between “***gg-hyd***” (located in Hyderabad) and “***gg-che***” (located in Chennai) from GRIDFS (located in Bangalore).



```
[henry@gridfs ~]$ globus-url-copy gsiftp://gg-hyd/tmp/weka.jar gsiftp://gg-che/tmp/weka.jar
[henry@gridfs ~]$
```

Fig. 4.11: globus-url-copy command – gsiftp to gsiftp

*In the above examples, the “**local machine”** refers to the machine from where the **globus-url-copy** command is invoked.*

GSIFTP

- *GSIFTP provides more reliable and high performance file transfer for Grid computing applications.*
- *User should have **valid proxy** before invoking gsiftp transfer.*
- ***globus-url-copy** is the command used for invoking gsiftp transfers*
- *GSIFTP can also be called as GridFTP.*

For more information about globus and its usage visit
http://globus.org

GARUDA Grid Meta Scheduler (Gridway):

GridWay enables you to treat your jobs as if they were Unix processes. Each job is given a numerical identifier, analogous to the PID of a process. This value is called the Job identifier, JID for short. If the job belongs to an array job, it will also have an array identifier, AID for short. A job index within an array is called the task identifier, TID for short.

Jobs are submitted using the **gwsubmit** command (Refer Fig. 4.14). A job is described by its template file. Here you can specify the job's executable file, its command line arguments, input/output files, standard stream redirection as well as other aspects.

Jobs can be monitored using the **gwps** command (Refer Fig. 4.15). You can control your jobs at runtime using the **gwkill** command. You can synchronize your jobs using the **gwwait** command. You can find out what resources your job has used with the **gwhistory** command.

System monitoring commands allow you to gather information of the GridWay system and the Grids you are using. These commands are: **gwuser** to show information about the users using GridWay; **gwhost** to monitor the available hosts in the testbed;

For more information about Gridway and its usage visit ***http://gridway.org***

What are the various states of my job?

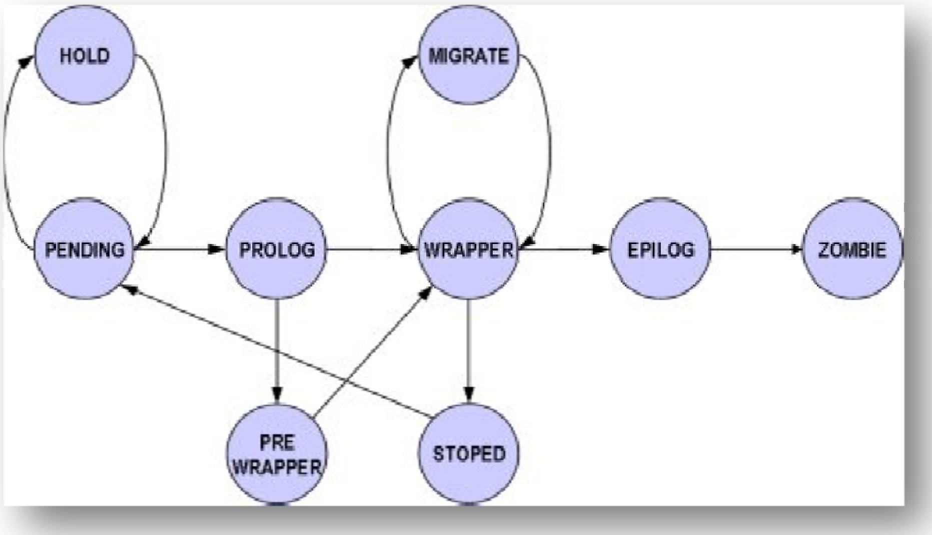


Fig. 4.12: Various states of a gridway job

A job can be in one of the following dispatch states (DM state):

- Pending (**pend**): The job is waiting for a resource to run on. The job reaches this state when it is initially submitted by the user or when it is restarted after a failure, stop or self-migration.
- Hold (**hold**): The owner (or GridWay administrator) has held the job. It will not be scheduled until it receives a release signal.
- Prolog (**prol**): The job is preparing the remote system, by creating the execution directory in the remote host and transferring the input and restart (in case of migration) files to it.
- Pre-wrapper (**prew**): The job is making some advanced preparation tasks in the remote resource, like getting some data from a service, obtaining software licenses, etc.

- Wrapper (**wrap**): The job is executing the Wrapper, which in turns executes the actual application. It also starts a self-monitoring program if specified. This monitor, watches the raw performance (CPU usage) obtained by the application.
- Epilog (**epil**): The job is finalizing. In this phase it transfers the output and restart (in case of failure, stop or self-migration) files and cleaning up the remote system directory.
- Migrate (**migr**): The job is migrating from one resource to another, by canceling the execution of Wrapper and performing finalization tasks in the old resource (like in Epilog state) and preparation tasks in the new resource (like in Prolog state).
- Stopped (**stop**): The job is stopped. If restart files have been defined in the Job Template, they are transferred back to the client, and will be used when the job is resumed.
- Failed (**fail**): The job failed.
- Done (**done**): The job is done and the user can check the exit status.

Overview of gridway job template

Syntax:

<VARIABLE> = ["]<VALUE>["]

E.g. EXECUTABLE = /bin/ls

Table 4.1 and 4.2 shows the sample job templates of a serial and parallel application respectively.

Job Template for serial program: Below is the sample gridway job template for serial program.

```
EXECUTABLE=/bin/ls
ARGUMENTS=-la file.txt
STDIN_FILE=/dev/null
STDOUT_FILE=stdout.${JOB_ID}
STDERR_FILE=stderr.${JOB_ID}
ENVIRONMENT=LD_LIBRARY_PATH=/usr/local/lib
REQUIREMENTS=HOSTNAME="gg-blr.tfg"
INPUT_FILES=file.txt
```

Table 4.1: Job template of a serial program

Job Template for parallel program: Below is the sample gridway job template for parallel program.

```
EXECUTABLE = /home/tfg/santhosh/test/pi
STDIN_FILE = /dev/null
STDOUT_FILE = ${JOB_ID}.out
STDERR_FILE = ${JOB_ID}.err
REQUIREMENTS = LRMS_TYPE != "Fork"
REQUIREMENTS=HOSTNAME="gg-blr.tfg"
TYPE = "mpi"
NP = 80
```

Table 4.2: Job template of a parallel program

Job template options overview

General:

Variable	Description
NAME	Name of the job (filename of the Job Template by default).

Execution:

Variable	Description
EXECUTABLE	The executable file. **Example:** **EXECUTABLE = bin.${ARCH}** **EXECUTABLE = /bin/ls** Executable file can be also be taken from GSRM, **EXECUTABLE =** ***gsiftp://gsrm.ctsf.cdac.org.in/dpm/ctsf.cdac.org.in/home/<VO_NAME>/<executable_file>***
ARGUMENTS	Arguments to the above executable. **Example:** **ARGUMENTS = "${TASK_ID}"**
ENVIRONMENT	User defined, comma-separated, environment variables. **Example:** **ENVIRONMENT = SCRATCH_DIR /tmp, LD_LIBRARY_PATH=/usr/local/lib**
TYPE	Type of job. Possible values are single (default), multiple and mpi, with similar behavior to that of GRAM jobs.
NP	Number of processors in MPI jobs. For multiple and single jobs it defines the count parameter in the RSL.

I/O files:

INPUT_FILES	A comma-separated pair of local remote filenames. If the remote filename is missing, the local filename will be preserved in the execution host. **Example:** **INPUT_FILES = param.${TASK_ID} param, inputfile** **INPUT_FILES = gsiftp://machine/tmp/input_exp1 input** **INPUT_FILES = file:///etc/passwd** **INPUT_FILES = test_case.bin** **INPUT_FILES =** ***gsiftp://gsrm.ctsf.cdac.org.in/dpm/ctsf.cdac.org.in/home/<VO_NAME>/input.txt*** **(Input files can be read from GSRM server given above)**
OUTPUT_FILES	**A comma-separated pair of remote filename local filename. If the local filename is missing, the remote filename will be preserved in the client host.** **Example:** **OUTPUT_FILES = output file, binary binary.${ARCH}.${TASK_ID}** **OUTPUT_FILES = output.bin /tmp/outfile** **GridFTP** ***URL*****: The file will be copied to the given GridFTP url** **OUTPUT_FILES = out gsiftp://storage_servere/~/output.${TASK_ID}** **OUTPUT_FILES = test_case.bin** **OUTPUT_FILES =** ***http://gsrm.ctsf.cdac.org.in/dpm/ctsf.cdac.org.in/home/<VO_NAME>/output.out*** **(Output files can be directly stored in GSRM server as given above)**

Standard streams:

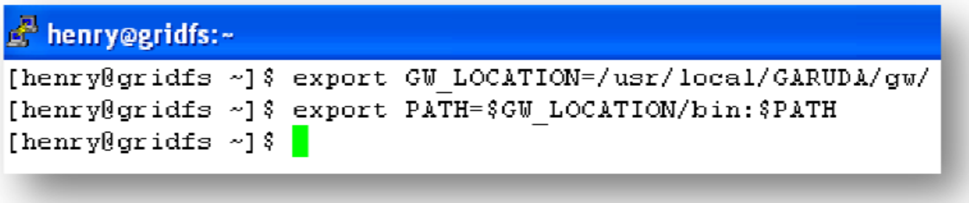
STDIN_FILE	Standard input file. **Example:** **STDIN_FILE = /dev/null** **STDIN_FILE = In/input.${JOB_ID}**
STDOUT_FILE	Standard output file. **Example:** **STDOUT_FILE = stdout_file.${JOB_ID}** **STDOUT_FILE = Out/ofile**
STDERR_FILE	Standard error file. **Example: STDERR_FILE = stderr_file.${JOB_ID}**

Resource selection:

REQUIREMENTS	A Boolean expression evaluated for each available host, if the evaluation returns true the host will be considered to submit the job. **Example:** **REQUIREMENTS = ARCH = “i686” & CPU_MHZ > 1000**
RANK	A numerical expression evaluated for each candidate host (those for which the requirement expression is true). Those candidates with higher ranks are used first to execute your jobs. **Example:** **RANK = (CPU_MHZ * 2) + FREE_MEM_MB**

How to set necessary environment variables to access gridway commands?

Fig. 4.13 depicts the necessary commands to set up the gridway user environment at gridfs machine. Note: The path of gridway installation may vary on different clusters.



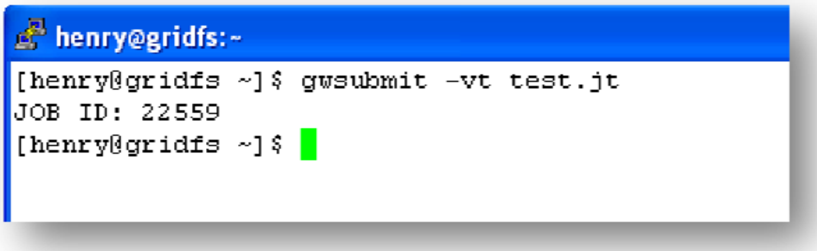
```
[henry@gridfs ~]$ export GW_LOCATION=/usr/local/GARUDA/gw/
[henry@gridfs ~]$ export PATH=$GW_LOCATION/bin:$PATH
[henry@gridfs ~]$ 
```

Fig. 4.13: Exporting gridway location

How to submit a job to gridway scheduler?

gwsubmit is the command used to submit jobs in gridway as shown in Fig. 4.14.

E.g. ***$ gwsubmit –vt <filename>***



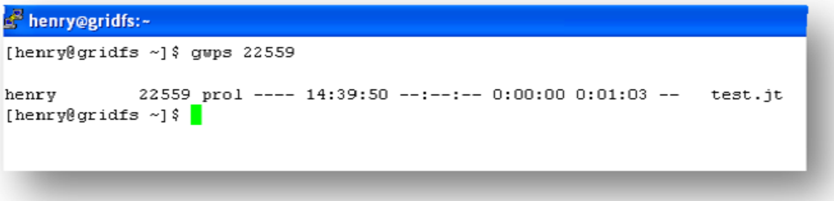
```
[henry@gridfs ~]$ gwsubmit -vt test.jt
JOB ID: 22559
[henry@gridfs ~]$ 
```

Fig. 4.14: gwsubmit command to submit jobs in gridway

How to check the status of my job?

As shown in Fig. 4.15, ***gwps*** is the command used to check the status of gridway job.

E.g. ***$ gwps <job_id>***

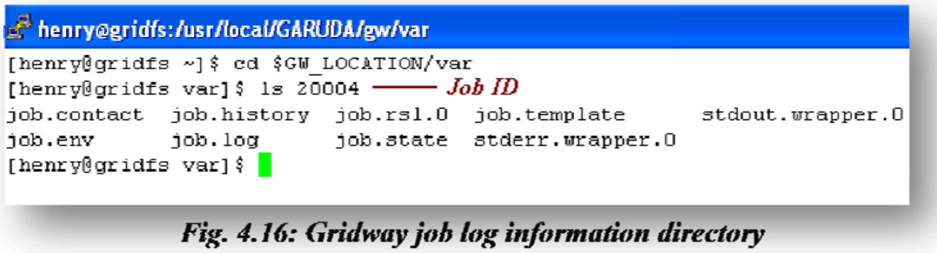


```
[henry@gridfs ~]$ gwps 22559

henry        22559 prol ---- 14:39:50 --:--:-- 0:00:00 0:01:03 --   test.jt
[henry@gridfs ~]$
```

Fig. 4.15: gwps command to check the status of the job

How will I troubleshoot my gridway job?

Debugging job execution: GridWay reporting and accounting facilities provide information about overall performance which helps to debug job execution. Gridway generates log files under the ***$GW_LOCATION/var*** directory:



```
[henry@gridfs ~]$ cd $GW_LOCATION/var
[henry@gridfs var]$ ls 20004
job.contact  job.history  job.rsl.0  job.template      stdout.wrapper.0
job.env      job.log      job.state  stderr.wrapper.0
[henry@gridfs var]$
```

Fig. 4.16: Gridway job log information directory

The below table explains each file in detail,

- **gwd.log**: System level log. You can find log information of the activity of the middleware access drivers; and coarse-grain log information about jobs.
- **$JOB_ID/job.log**: Detailed log information for each job, it includes details of job state transitions, resource usage and performance.
- **$JOB_ID/stdout.wrapper**: Standard output of the wrapper executable.
- **$JOB_ID/stderr.wrapper**: Standard error output of the wrapper executable. By default, wrapper is executed with shell debugging options (-xv) active, so this is usually the best source of information in case of failure.

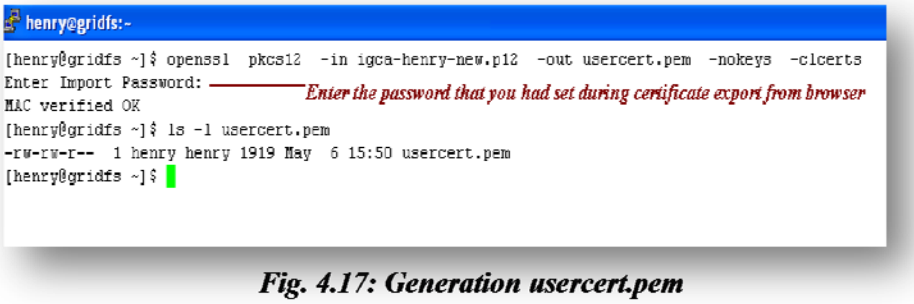
IGCA Command Reference:

For Linux Users:

How to convert .p12 or .pfx files to .pem?

Assume the file name as "***igca-henry-new.p12***". As shown in Fig. 4.17, type the below command to extract usercert.pem.

openssl pkcs12 -in igca-henry-new.p12 -out usercert.pem -nokeys -clcerts



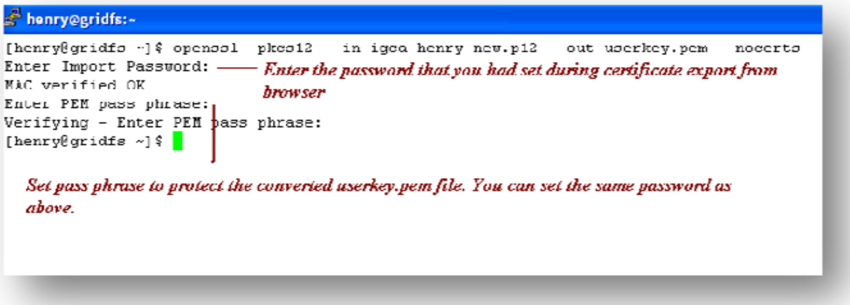
```
[henry@gridfs ~]$ openssl  pkcs12  -in igca-henry-new.p12  -out usercert.pem  -nokeys  -clcerts
Enter Import Password:
MAC verified OK
[henry@gridfs ~]$ ls -l usercert.pem
-rw-rw-r--  1 henry henry 1919 May  6 15:50 usercert.pem
[henry@gridfs ~]$
```

Enter the password that you had set during certificate export from browser

Fig. 4.17: Generation usercert.pem

As shown in Fig. 4.18, type the below command to extract userkey.pem.

openssl pkcs12 -in igca-henry-new.p12 -out userkey.pem -nocerts



```
[henry@gridfs ~]$ openssl  pkcs12   in igca henry new.p12   out userkey.pem   nocerts
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
[henry@gridfs ~]$
```

Enter the password that you had set during certificate export from browser

Set pass phrase to protect the converted userkey.pem file. You can set the same password as above.

Fig. 4.18: Generating userkey.pem

For Windows Users:

If you are a windows user, then follow the below steps,

1. Download and Install OpenSSL for Windows. (***http://www.slproweb.com/download/Win32OpenSSL_Light-0_9_8r.exe***)
2. Install OpenSSL into the folder ***C:\OpenSSL***
3. As shown in Fig 4.19, place your ***.pfx or .p12*** files in the folder ***C:\my-cert***

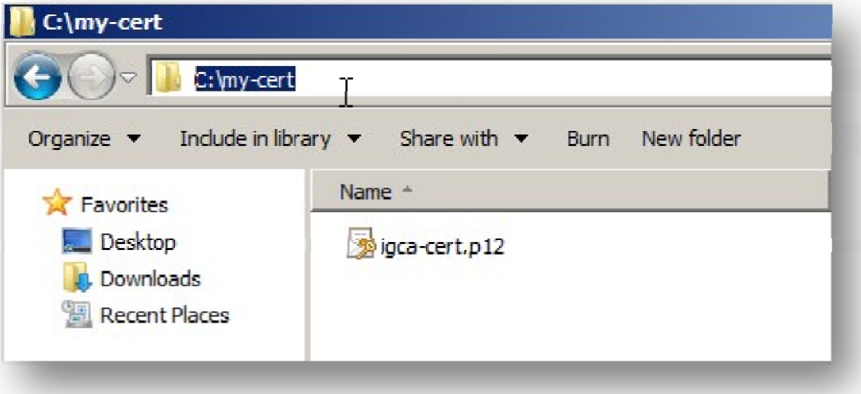
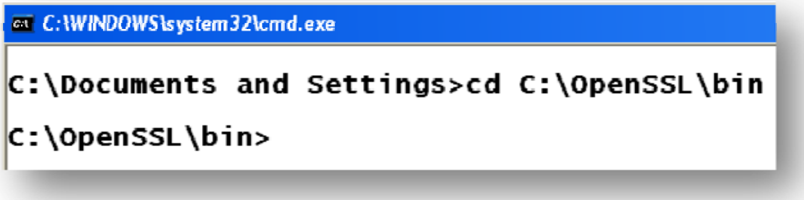


Fig 4.19 Place your .p12 or .pfx files in c:\my-cert

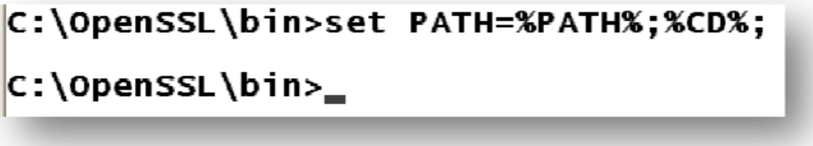
4. As shown in Fig 4.20, open your command prompt and enter the folder ***C:\OpenSSL\bin***



```
C:\Documents and Settings>cd C:\OpenSSL\bin
C:\OpenSSL\bin>
```

Fig 4.20: Change directory to openssl installed folder.

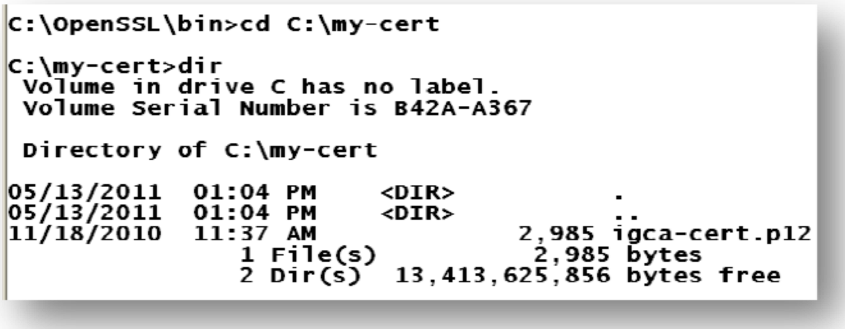
5. Set the OpenSSL commands in the ***PATH*** as shown in Fig. 4.21



```
C:\OpenSSL\bin>set PATH=%PATH%;%CD%;
C:\OpenSSL\bin>_
```

Fig 4.21: Set the OpenSSL command in the PATH

6. Change to the directory ***c:\my-cert***, where your IGCA certificate (.p12) is stored.



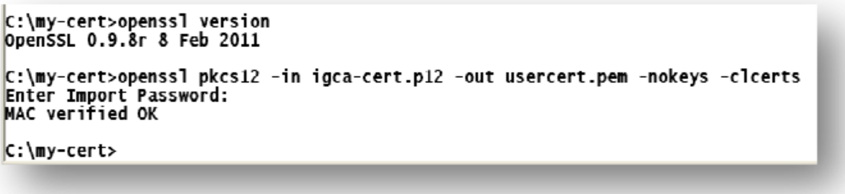
```
C:\OpenSSL\bin>cd C:\my-cert

C:\my-cert>dir
 Volume in drive C has no label.
 Volume Serial Number is B42A-A367

 Directory of C:\my-cert

05/13/2011  01:04 PM    <DIR>          .
05/13/2011  01:04 PM    <DIR>          ..
11/18/2010  11:37 AM             2,985 igca-cert.p12
               1 File(s)          2,985 bytes
               2 Dir(s)  13,413,625,856 bytes free
```

Fig 4.22: Change to the .p12 certificate directory

7. **Extracting usercert.pem**: As shown in Fig. 4.23, type the below command to extract ***usercert.pem***.

   openssl pkcs12 –in igca-cert.p12 –out usercert.pem –nokeys –clcerts

   When prompted for "***Import Password***", enter the password which you had set during your certificate export from the browser.



```
C:\my-cert>openssl version
OpenSSL 0.9.8r 8 Feb 2011

C:\my-cert>openssl pkcs12 -in igca-cert.p12 -out usercert.pem -nokeys -clcerts
Enter Import Password:
MAC verified OK

C:\my-cert>
```

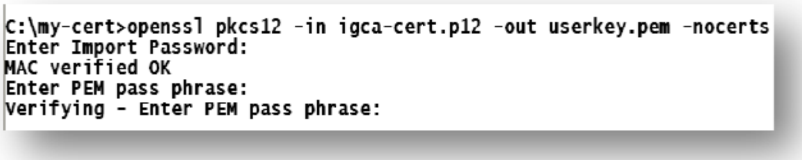
Fig 4.23: Generate usercert.pem

8. **Extracting userkey.pem:** As shown in Fig: 4.24, type the below command to extract userkey.pem

   openssl pkcs12 –in igca-cert.p12 –out userkey.pem –nocerts

When prompted for “***Import Password***”, enter the password which you had set during your certificate export from the browser.

When prompted for “***Enter PEM pass phrase***”, set new password to protect your userkey.pem file. Note: You can also enter the same password which you had set during your certificate export from the browser



```
C:\my-cert>openssl pkcs12 -in igca-cert.p12 -out userkey.pem -nocerts
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Fig 4.24: Generating userkey.pem

What is .p12 or .pfx file?

- ***.p12 or .pfx*** *– Certificate format that contains both public and private key.*
- *This file is normally protected with a password.*

What is .pem file?

- *Certificate format used in grid.*
- ***usercert.pem*** *–public key.*
- ***userkey.pem*** *– private key.*

Chapter 5

GARUDA GridFTP Client

Chapter 5: GARUDA GridFTP Client

What is GARUDA GridFTP client?

Garuda GridFTP Client is a GUI for performing file transfers using the Grid FTP protocol. It is intended to be an easy-to-use program for remote file operations such as upload/download, directory and file listing and file delete. The software is available in versions that will run on many different flavors of the Linux operating system, as well as on 32-bit flavors of Microsoft Windows (XP, Vista and 7).

What are the Pre-requisites?

You need Java (at least 1.6) to run this software. Also you need Garuda certificates properly setup on your machine, to connect to the remote resources of the grid. Please refer to ***http://portal.garudaindia.in/gridftp-gui*** for a detailed list of pre-requisites

Where can I find online documentation for this tool?

Online Documentation is maintained at ***http://portal.garudaindia.in/gridftp-gui***

GridFTP GUI Usage:

For Windows Users:

1. Download the Garuda Gridftp software for windows from the below link: ***http://gridfs.ctsf.cdac.org.in/gridftp-gui/download/Garuda-GridFTP-Setup.exe***
2. Install the Garuda GridFTP software by just double clicking the setup file.
3. Once installed you can invoke the program using: ***Start Menu Item -> Garuda -> Grid FTP GUI*** as shown in Fig. 5.1.

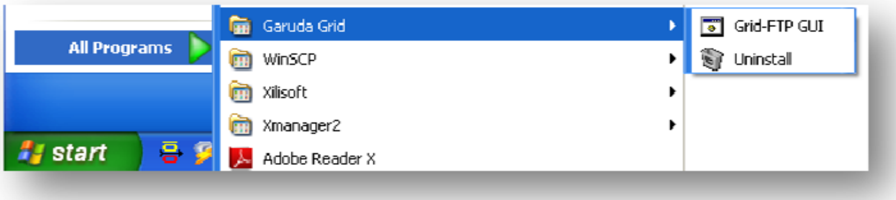
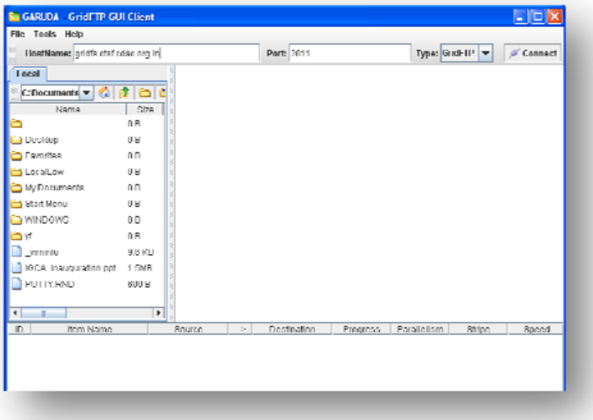


Fig. 5.1: Invoking GridFTP GUI software

4. Step 3 will open a GUI window as shown in Fig. 5.2.



HostName: gridfs.ctsf.cdac.org.in

Port: 2811

Type: GridFTP

Fig. 5.2: GridFTP GUI window

5. Upload your certificate to GridFTP client by clicking ***Tools -> Options*** as shown in Fig. 5.3.

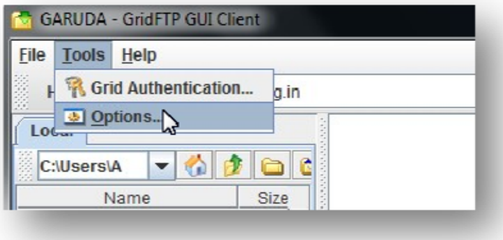


Fig. 5.3: Click Tools > Options

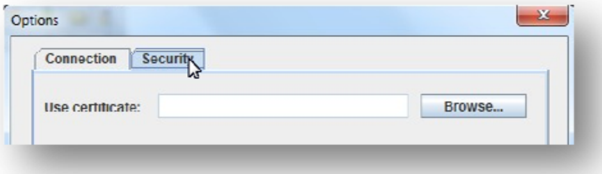
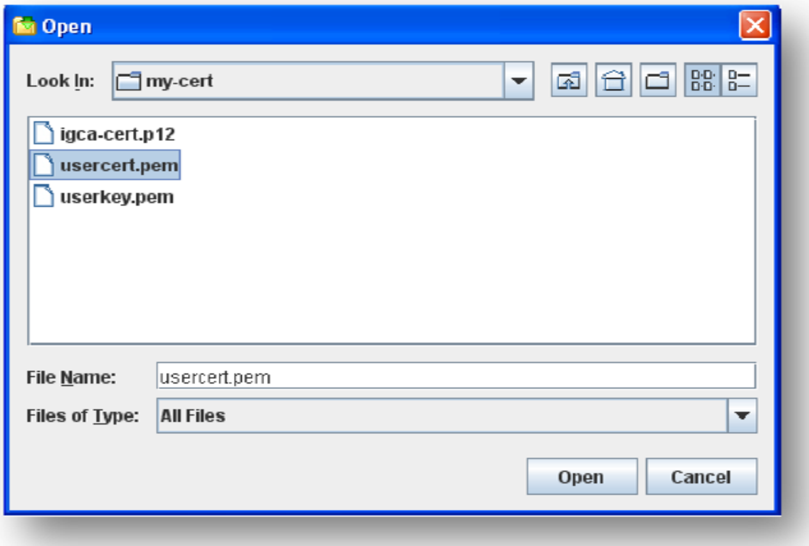


Fig. 5.4: Click "Security" tab in Options dialog box.

6. Click the "***Browse***" to upload your certificate (***usercert.pem***) to the client GUI as shown in Fig. 5.4 & Fig 5.5:



File Name: usercert.pem

Files of Type: All Files

Fig. 5.5: Upload usercert.pem to the client tool

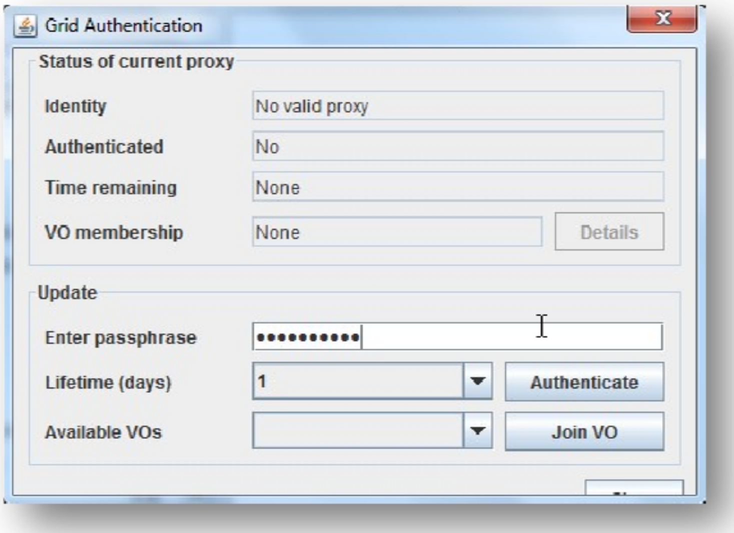
What if I don't have a .pem certificate file?

***Sol**: Refer Chapter 4: IGCA command reference to convert your ".p12" certificate file to ".pem" certificate file.*

How to get my .p12 file?

Sol:** You need to export your certificate from the web browser. Refer the below link for more information,* ***http://ca.garudaindia.in/index.php/information/faq/export-the-certificates-obtain-pfxps12-file/

7. Click Tools -> Grid Authentication to authenticate with the GridFTP server. Enter the passphrase and click "Authenticate" button as shown in Fig. 5.6.



Grid Authentication

Status of current proxy

Identity: No valid proxy

Authenticated: No

Time remaining: None

VO membership: None

Update

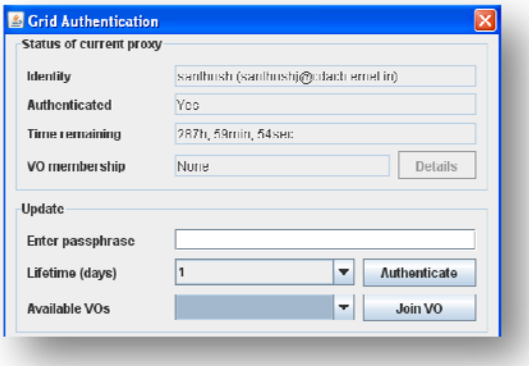
Enter passphrase: ••••••••••

Lifetime (days): 1

Available VOs:

Fig. 5.6: Authentication to the GridFTP server

8. Upon successful authentication, you should be able to view the status of your proxy as shown in Fig. 5.7.



Status of current proxy

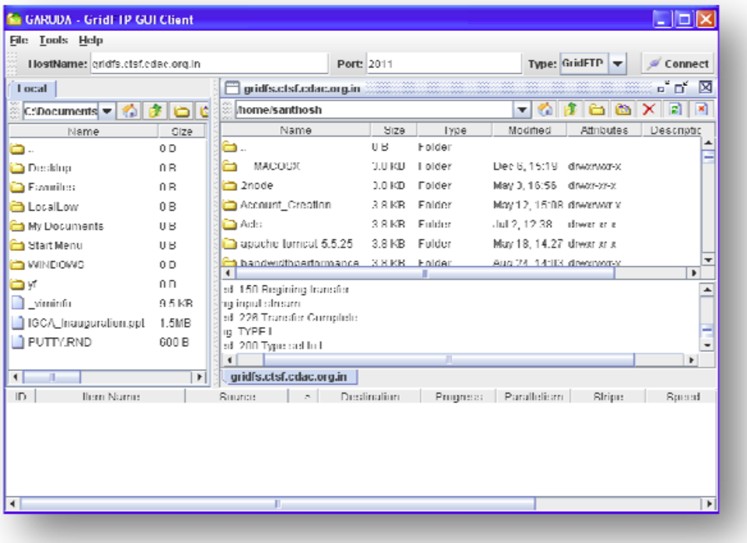
Identity	santhosh (santhoshj@cdac.ernet.in)
Authenticated	Yes
Time remaining	287h, 59min, 54sec
VO membership	None

Update

Enter passphrase	
Lifetime (days)	1
Available VOs	

Fig. 5.7: View the status of your proxy

9. Connect to the GridFTP Server with ***Hostname=”gridfs.ctsf.cdac.org.in”, port=”2811”, Type=”GridFTP”*** and click “***Connect***” button. Upon successful connection, you should be able to view a transfer window as shown in Fig. 5.8.



HostName: gridfs.ctsf.cdac.org.in Port: 2811 Type: GridFTP

C:\Documents

Name	Size
..	0 B
Desktop	0 B
Favorites	0 B
LocalLow	0 B
My Documents	0 B
Start Menu	0 B
WINDOWS	0 B
yf	0 B
_viminfo	9.5 KB
IGCA_Inauguration.ppt	1.5MB
PUTTY.RND	600 B

/home/santhosh

Name	Size	Type	Modified	Attributes	Descriptio
..	0 B	Folder			
MACOSX	3.0 KB	Folder	Dec 6, 15:19	drwxrwxr-x	
2node	3.0 KB	Folder	May 3, 16:56	drwxr-xr-x	
Account_Creation	3.8 KB	Folder	May 12, 15:08	drwxrwxr-x	
Ads	3.8 KB	Folder	Jul 2, 12:38	drwxr-xr-x	
apache-tomcat-5.5.25	3.8 KB	Folder	May 18, 14:27	drwxr-xr-x	
bandwidthperformance	3.8 KB	Folder	Aug 24, 14:03	drwxrwxr-x	

```
ed 150 Beginning transfer
ng input stream
ed 226 Transfer Complete
ng TYPE I
ed 200 Type set to I
```

ID	Item Name	Source	^	Destination	Progress	Parallelism	Stripe	Speed

Fig. 5.8: Connected to GridFTP Server

For Linux users:

If you are a Linux user, then follow the below steps,

a. Download the Zip file from ***http://portal.garudaindia.in/gridftp-gui/download/garuda-gridftp.zip***
b. Extract the contents of the download zip file. The extracted folder will be the TOOL_HOME directory. Set Java to be available on the PATH.
c. The extracted content will contain a folder called '***certificates***'. This contains the CA certificate files.
d. Login as root and move this folder to **/etc/grid-security/.** If the '***grid-security***' folder is not present in ***/etc***, then please create it.
e. Use ***run.sh*** script available in the TOOL_HOME folder, to invoke the software.
f. Refer the Fig. 5.2 to Fig 5.8 for more information tool usage.

Where can I find, online documentation for this tool?

Garuda GridFTP Client manual can be found from the below URL.
http://portal.garudaindia.in/gridftp-gui/gridftp-gui.htm

How to access GridFTP in command line?

Refer Chapter 4.

Chapter 6

Garuda Storage Resource Manager (GSRM)

Chapter 6: Garuda Storage Resource Manager (GSRM)

What is GSRM?

GARUDA **S**torage **R**esource **M**anager (GSRM) is a peer to peer data grid solution for GARUDA. An SRM can be defined as a middle ware component that manages the dynamic use of storage resources in the grid. GSRM provides,

- **File and Directory managemen**t – Upload, download of files and directories.
- **Dynamic space management**: Reserving & release of storage spaces and setting lifetime for the files.
- **User friendly interfaces** like command line, APIs and web interface.
- **Interfacing with the Compute Grid:** Allows users to transfer their required data from compute grid to GSRM and vice versa.

What are the steps to access GSRM?

Below are the steps to access GSRM services,

a) Login to GRIDFS machine. Refer Chapter 3.
b) Generate valid credentials with VO attributes.
c) Setting up of GSRM user environment.
d) Access GSRM via GSRM client commands.

What are the GSRM end points in GARUDA?

GSRM end points are referred as GSRM servers. Below are the GSRM end points in GARUDA,

Location	End point URL	Hostname
Bangalore	httpg:// gsrm.ctsf.cdac.org.in:8446/	gsrm
Hyderabad	httpg:// srmhyd.cdac.in:8446/	srmhyd

Table 6.1: GSRM End points in GARUDA

Note: End points use protocol ***httpg***

What is SURL?

A file or a directory located in a GSRM storage space is identified by a Site URL (**SURL**). Below is the format of SURL,

srm://host[:port]/StFN root path/site_file_name

An example of SURL is:

srm://gsrm:8446/dpm/ctsf.cdac.org.in/home/GarudaAdmin/weka.jar

What is StFN?

StFN *stands for "**St**orage **F**ile **N**ame". **SFN** is a part of SURL.*

What is my StFN root path?

*Below is the general format of StFN for **gsrm.ctsf.cdac.org.in** ,*
/dpm/ctsf.cdac.org.in/home/<VO_NAME>
Below is the general format of SFN for ***srmhyd.cdac.in***,
/dpm/cdac.in/home/<VO_NAME>

*E.g. consider a user who belongs to "**OSDD**" VO. His **StFN** for gsrm.ctsf.cdac.org.in is **/dpm/ctsf.cdac.org.in/home/OSDD***

*All the GSRM examples in this book are of a user who belongs to "**GarudaAdmin**" VO. So the StFN will look like "**/dpm/ctsf.cdac.org.in/home/GarudaAdmin**". Replace this with your corresponding StFN.*

Setting up GSRM User Environment:

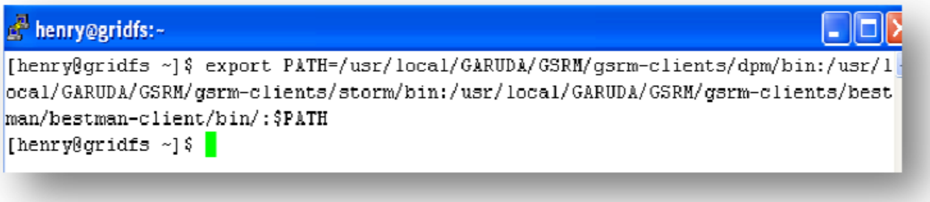
GSRM Client commands are located at GRIDFS machine as shown below Table 6.2.

Clients	Location
storm	*/usr/local/GARUDA/GSRM/gsrm-clients/storm/bin*
dpm	*/usr/local/GARUDA/GSRM/gsrm-clients/dpm/bin*
bestman	*/usr/local/GARUDA/GSRM/gsrm-clients/bestman/bestman-client/bin/*

Table 6.2: Location of GSRM clients

Setting up GSRM client commands in "PATH" variable:

Fig. 6.1 shows, setting up of GSRM client commands in "PATH" environment variable.

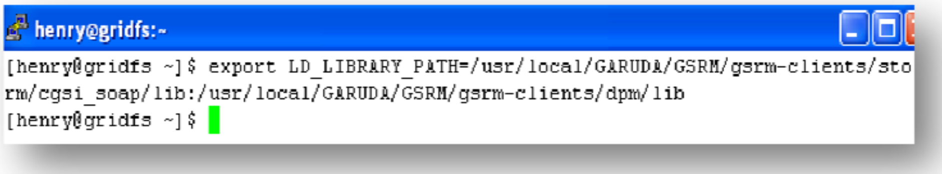


```
[henry@gridfs ~]$ export PATH=/usr/local/GARUDA/GSRM/gsrm-clients/dpm/bin:/usr/l
ocal/GARUDA/GSRM/gsrm-clients/storm/bin:/usr/local/GARUDA/GSRM/gsrm-clients/best
man/bestman-client/bin/:$PATH
[henry@gridfs ~]$
```

Fig. 6.1: Setting up a GSRM client environment

Setting up of GSRM Libraries in LD_LIBRARY_PATH:

As shown in Fig. 6.2, export LD**_LIBRARY_PATH** to locate GSRM libraries.



```
[henry@gridfs ~]$ export LD_LIBRARY_PATH=/usr/local/GARUDA/GSRM/gsrm-clients/sto
rm/cgsi_soap/lib:/usr/local/GARUDA/GSRM/gsrm-clients/dpm/lib
[henry@gridfs ~]$
```

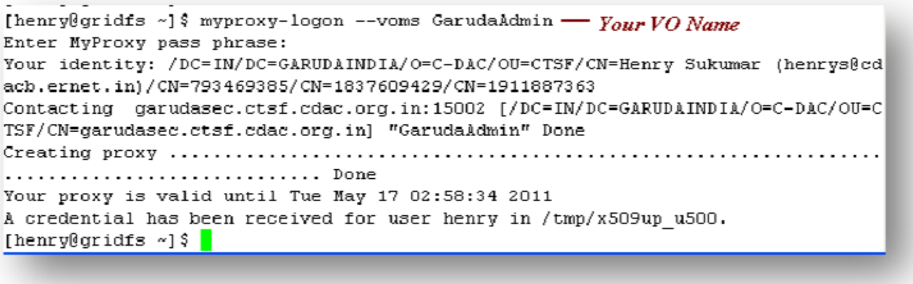
Fig. 6.2 Setting LD_LIBRARY_PATH

How to download credentials with VO attributes?

In order to access GSRM services, user needs to have a valid VO credentials. Fig. 6.3 shows how to download credentials from MyProxy repository and add VO attributes to it.

Command:
myproxy-logon --voms <VO_NAME>

E.g. Below example generates credentials for VO called “GarudaAdmin”; you need to replace this with your VO name.



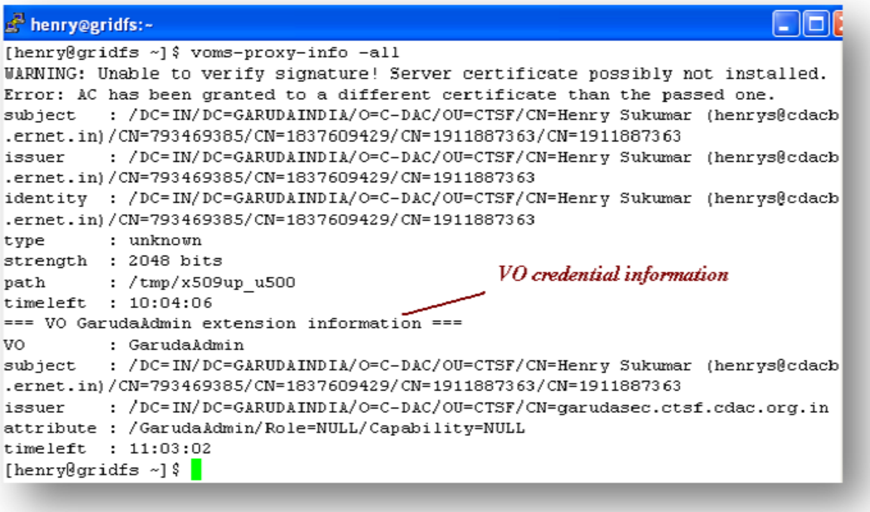
```
[henry@gridfs ~]$ myproxy-logon --voms GarudaAdmin
Enter MyProxy pass phrase:
Your identity: /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cd
acb.ernet.in)/CN=793469385/CN=1837609429/CN=1911887363
Contacting  garudasec.ctsf.cdac.org.in:15002 [/DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=C
TSF/CN=garudasec.ctsf.cdac.org.in] "GarudaAdmin" Done
Creating proxy .......................................................
............................. Done
Your proxy is valid until Tue May 17 02:58:34 2011
A credential has been received for user henry in /tmp/x509up_u500.
[henry@gridfs ~]$
```

Fig.: 6.3 Generating VO credentials

If you are not sure, which VO you belong to?
Send a mail to voms@cdacb.ernet.in

How to view my VO credentials?

As shown in Fig. 6.4, ***voms-proxy-info*** is the command used to view your credential information.



```
[henry@gridfs ~]$ voms-proxy-info -all
WARNING: Unable to verify signature! Server certificate possibly not installed.
Error: AC has been granted to a different certificate than the passed one.
subject   : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cdacb
.ernet.in)/CN=793469385/CN=1837609429/CN=1911887363/CN=1911887363
issuer    : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cdacb
.ernet.in)/CN=793469385/CN=1837609429/CN=1911887363
identity  : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cdacb
.ernet.in)/CN=793469385/CN=1837609429/CN=1911887363
type      : unknown
strength  : 2048 bits
path      : /tmp/x509up_u500
timeleft  : 10:04:06
=== VO GarudaAdmin extension information ===
VO        : GarudaAdmin
subject   : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cdacb
.ernet.in)/CN=793469385/CN=1837609429/CN=1911887363/CN=1911887363
issuer    : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=garudasec.ctsf.cdac.org.in
attribute : /GarudaAdmin/Role=NULL/Capability=NULL
timeleft  : 11:03:02
[henry@gridfs ~]$
```

Fig. 6.4: voms-proxy-info command shows VO credentials information

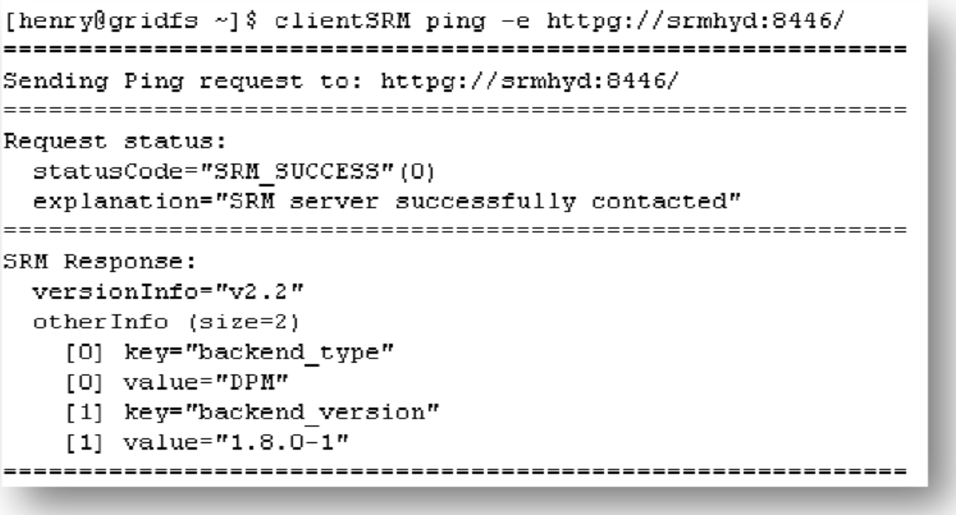
How to check the status of GSRM server?

"clientSRM" command along with "***ping***" function is used to check the state of the SRM and it works as an "Are you alive" type of call. The version of the SRM specification implemented by the server is returned.

As shown Fig. 6.5, "***ping***" function of the ***clientSRM*** command is used to ping a GSRM server (gsrm and srmhyd):

clientSRM ping -e httpg://gsrm.ctsf.cdac.org.in:8446/

clientSRM ping -e httpg://srmhyd:8446/



```
[henry@gridfs ~]$ clientSRM ping -e httpg://srmhyd:8446/
============================================================
Sending Ping request to: httpg://srmhyd:8446/
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
  explanation="SRM server successfully contacted"
============================================================
SRM Response:
  versionInfo="v2.2"
  otherInfo (size=2)
    [0] key="backend_type"
    [0] value="DPM"
    [1] key="backend_version"
    [1] value="1.8.0-1"
============================================================
```

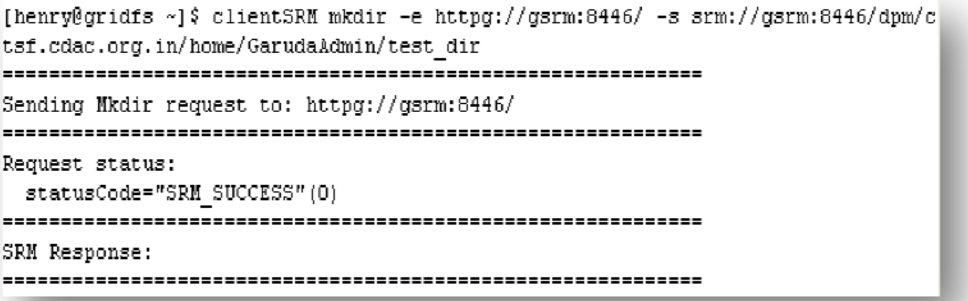
Fig. 6.5: clientSRM command is used to ping the GSRM server

Note: Type “*clientSRM ping -help*” for more information.

How to create a directory in GSRM?

As shown in Fig. 6.6, “***mkdir***” function of the ***clientSRM*** command is used to create directory in GSRM server. Type “***clientSRM mkdir –help***” for more information.

clientSRM mkdir –e <GSRM_End_Point> -s <SURL>



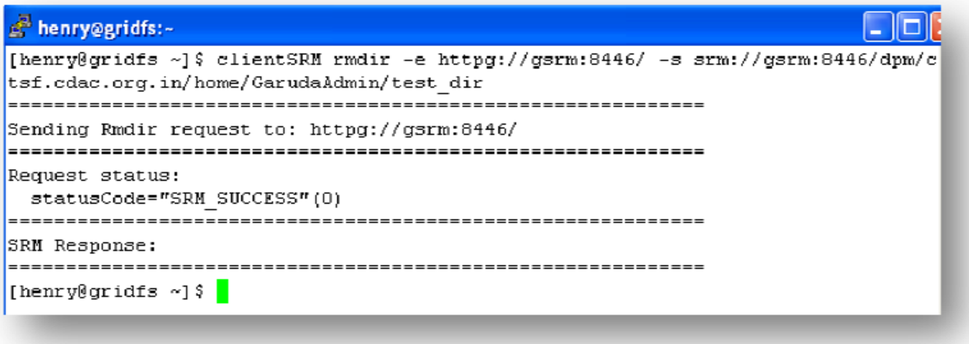
```
[henry@gridfs ~]$ clientSRM mkdir -e httpg://gsrm:8446/ -s srm://gsrm:8446/dpm/c
tsf.cdac.org.in/home/GarudaAdmin/test_dir
============================================================
Sending Mkdir request to: httpg://gsrm:8446/
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
============================================================
```

Fig. 6.6: Creating directory in GSRM server

How to remove a directory in GSRM?

As shown in Fig. 6.7, "***rmdir***" function of the ***clientSRM*** command is used to the remove directory in GSRM server.

clientSRM rmdir -e <GSRM_End_Point> -s <SURL>



```
[henry@gridfs ~]$ clientSRM rmdir -e httpg://gsrm:8446/ -s srm://gsrm:8446/dpm/c
tsf.cdac.org.in/home/GarudaAdmin/test_dir
============================================================
Sending Rmdir request to: httpg://gsrm:8446/
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
============================================================
[henry@gridfs ~]$ 
```

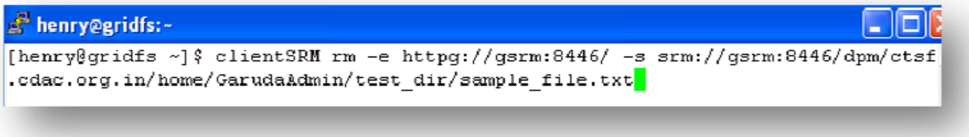
Fig. 6.7 Remove the directory in GSRM

Note: The above command is used to remove empty directory. Use "-r" option to remove the directory recursively. **Type "*clientSRM rmdir –help*" for more information.**

How to remove a file in GSRM?

As shown in Fig. 6.8, "***rm***" function of the ***clientSRM*** command is used to remove a specific file.

clientSRM rm -e <GSRM_End_Point> -s <SURL>



```
[henry@gridfs ~]$ clientSRM rm -e httpg://gsrm:8446/ -s srm://gsrm:8446/dpm/ctsf
.cdac.org.in/home/GarudaAdmin/test_dir/sample_file.txt
```

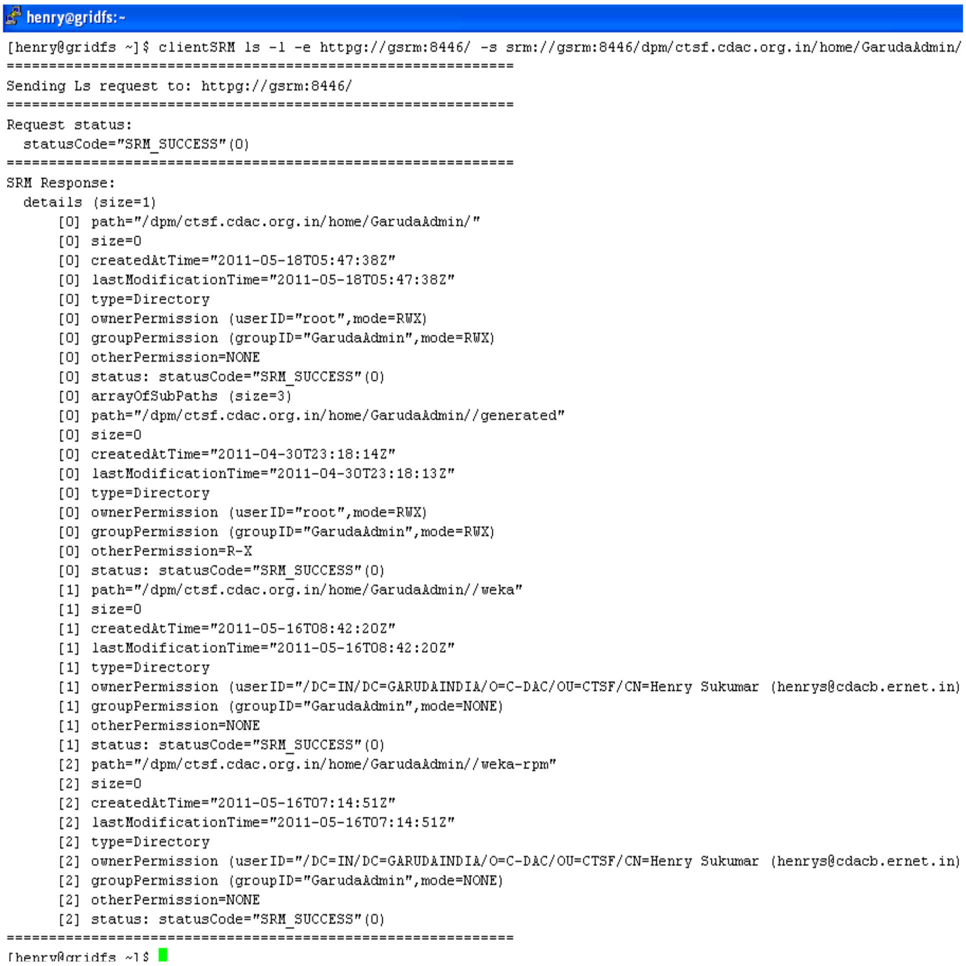
Fig. 6.8: Removing a specific file in a GSRM server

Note: Type "***clientSRM rm –help***" for more information.

How to list the files in a directory?

As shown in Fig. 6.9, “***ls***” function of the ***clientSRM*** command is used to list the files in a directory.

clientSRM ls -l -e <GSRM_End_Point> -s <SURL>



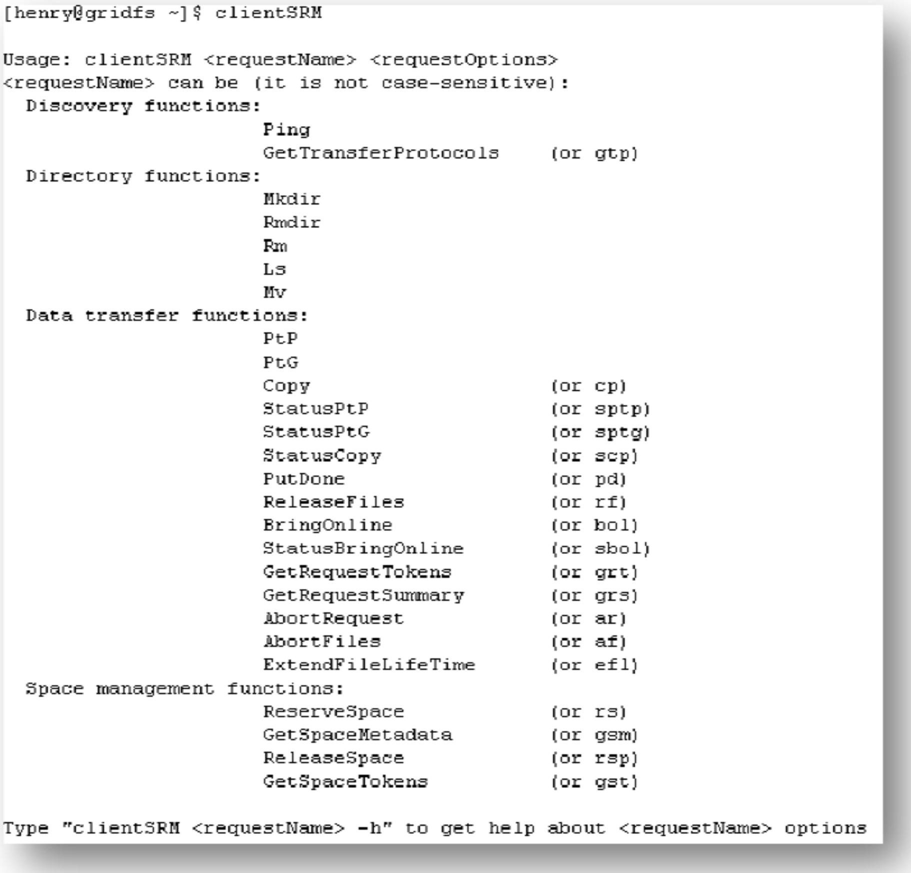
```
[henry@gridfs ~]$ clientSRM ls -l -e httpg://gsrm:8446/ -s srm://gsrm:8446/dpm/ctsf.cdac.org.in/home/GarudaAdmin/
============================================================
Sending Ls request to: httpg://gsrm:8446/
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
  details (size=1)
      [0] path="/dpm/ctsf.cdac.org.in/home/GarudaAdmin/"
      [0] size=0
      [0] createdAtTime="2011-05-18T05:47:38Z"
      [0] lastModificationTime="2011-05-18T05:47:38Z"
      [0] type=Directory
      [0] ownerPermission (userID="root",mode=RWX)
      [0] groupPermission (groupID="GarudaAdmin",mode=RWX)
      [0] otherPermission=NONE
      [0] status: statusCode="SRM_SUCCESS"(0)
      [0] arrayOfSubPaths (size=3)
      [0] path="/dpm/ctsf.cdac.org.in/home/GarudaAdmin//generated"
      [0] size=0
      [0] createdAtTime="2011-04-30T23:18:14Z"
      [0] lastModificationTime="2011-04-30T23:18:13Z"
      [0] type=Directory
      [0] ownerPermission (userID="root",mode=RWX)
      [0] groupPermission (groupID="GarudaAdmin",mode=RWX)
      [0] otherPermission=R-X
      [0] status: statusCode="SRM_SUCCESS"(0)
      [1] path="/dpm/ctsf.cdac.org.in/home/GarudaAdmin//weka"
      [1] size=0
      [1] createdAtTime="2011-05-16T08:42:20Z"
      [1] lastModificationTime="2011-05-16T08:42:20Z"
      [1] type=Directory
      [1] ownerPermission (userID="/DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cdacb.ernet.in)
      [1] groupPermission (groupID="GarudaAdmin",mode=NONE)
      [1] otherPermission=NONE
      [1] status: statusCode="SRM_SUCCESS"(0)
      [2] path="/dpm/ctsf.cdac.org.in/home/GarudaAdmin//weka-rpm"
      [2] size=0
      [2] createdAtTime="2011-05-16T07:14:51Z"
      [2] lastModificationTime="2011-05-16T07:14:51Z"
      [2] type=Directory
      [2] ownerPermission (userID="/DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar (henrys@cdacb.ernet.in)
      [2] groupPermission (groupID="GarudaAdmin",mode=NONE)
      [2] otherPermission=NONE
      [2] status: statusCode="SRM_SUCCESS"(0)
============================================================
[henry@gridfs ~]$
```

Fig. 6.9: listing the files in a directory

Note: Type “***clientSRM ls –help***” for more information.

What are the various functions that can be performed using clientSRM command?

"clientSRM" is one of the SRM client commands that can be used to perform various storage operations. As shown in Fig. 6.10, type "***clientSRM***" to view the help page.



```
[henry@gridfs ~]$ clientSRM

Usage: clientSRM <requestName> <requestOptions>
<requestName> can be (it is not case-sensitive):
  Discovery functions:
                      Ping
                      GetTransferProtocols     (or gtp)
  Directory functions:
                      Mkdir
                      Rmdir
                      Rm
                      Ls
                      Mv
  Data transfer functions:
                      PtP
                      PtG
                      Copy                     (or cp)
                      StatusPtP                (or sptp)
                      StatusPtG                (or sptg)
                      StatusCopy               (or scp)
                      PutDone                  (or pd)
                      ReleaseFiles             (or rf)
                      BringOnline              (or bol)
                      StatusBringOnline        (or sbol)
                      GetRequestTokens         (or grt)
                      GetRequestSummary        (or grs)
                      AbortRequest             (or ar)
                      AbortFiles               (or af)
                      ExtendFileLifeTime       (or efl)
  Space management functions:
                      ReserveSpace             (or rs)
                      GetSpaceMetadata         (or gsm)
                      ReleaseSpace             (or rsp)
                      GetSpaceTokens           (or gst)

Type "clientSRM <requestName> -h" to get help about <requestName> options
```

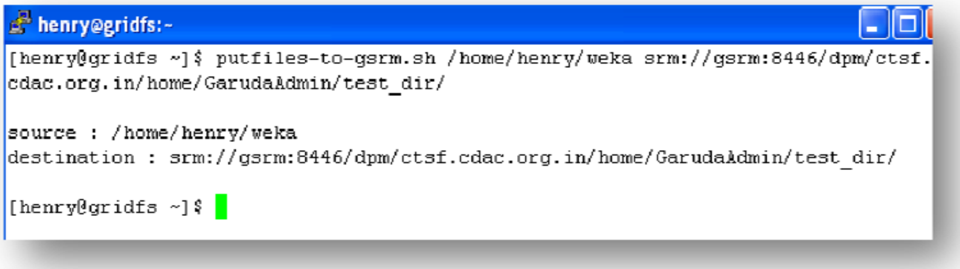
Fig. 6.10: clientSRM command and its various options

Note: Type "***clientSRM <function> -help***" to view detailed options for a particular function.

How to upload a directory to GSRM?

As shown in Fig. 6.11, "***putfiles-to-gsrm.sh***" is the command used to upload a directory to GSRM server.

Command: *putfiles-to-gsrm.sh* ***<Local_source_path> <SURL>***



```
[henry@gridfs ~]$ putfiles-to-gsrm.sh /home/henry/weka srm://gsrm:8446/dpm/ctsf.
cdac.org.in/home/GarudaAdmin/test_dir/

source : /home/henry/weka
destination : srm://gsrm:8446/dpm/ctsf.cdac.org.in/home/GarudaAdmin/test_dir/

[henry@gridfs ~]$
```

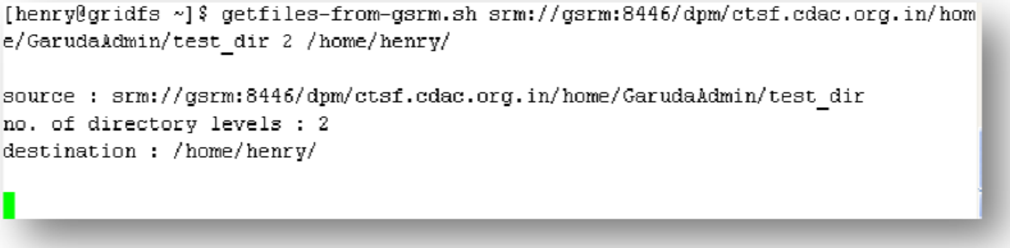
Fig. 6.11: Uploading a directory to GSRM

How to download a directory from GSRM?

As shown in Fig. 6.12, "***getfiles-from-gsrm.sh***" is the command used to download a directory from GSRM server.

Command:

getfiles-from-gsrm.sh ***<SURL> <directory_level> <Local_destination_path>***



```
[henry@gridfs ~]$ getfiles-from-gsrm.sh srm://gsrm:8446/dpm/ctsf.cdac.org.in/hom
e/GarudaAdmin/test_dir 2 /home/henry/

source : srm://gsrm:8446/dpm/ctsf.cdac.org.in/home/GarudaAdmin/test_dir
no. of directory levels : 2
destination : /home/henry/
```

Fig. 6.12: Downloading a directory from GSRM

How to upload a file to GSRM using clientSRM command?

Uploading a file to GSRM storage server involves 3 steps,

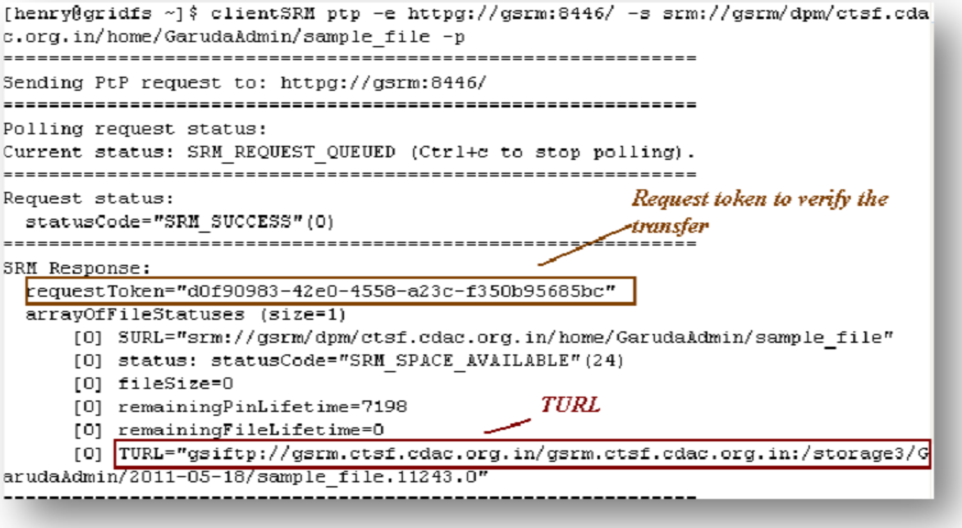
Step 1: Prepare to put request to GSRM. (***clientSRM ptp***)
Step 2: Transfer the file. (***globus-url-copy***)
Step 3: Verify the transfer. (***clientSRM pd***)

Step 1: Prepare to put request to GSRM

As shown in Fig. 6.13, ***clientSRM*** command with "***prepare to put request (ptp)***" function is used to make a put request to transfer a file named "***sample_file***" to Bangalore GSRM server (gsrm).

Command:

clientSRM ptp -e httpg://gsrm:8446/ -s srm://gsrm/dpm/ctsf.cdac.org.in/home/GarudaAdmin/sample_file –p



```
[henry@gridfs ~]$ clientSRM ptp -e httpg://gsrm:8446/ -s srm://gsrm/dpm/ctsf.cda
c.org.in/home/GarudaAdmin/sample_file -p
============================================================
Sending PtP request to: httpg://gsrm:8446/
============================================================
Polling request status:
Current status: SRM_REQUEST_QUEUED (Ctrl+c to stop polling).
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
  requestToken="d0f90983-42e0-4558-a23c-f350b95685bc"
  arrayOfFileStatuses (size=1)
      [0] SURL="srm://gsrm/dpm/ctsf.cdac.org.in/home/GarudaAdmin/sample_file"
      [0] status: statusCode="SRM_SPACE_AVAILABLE"(24)
      [0] fileSize=0
      [0] remainingPinLifetime=7198
      [0] remainingFileLifetime=0
      [0] TURL="gsiftp://gsrm.ctsf.cdac.org.in/gsrm.ctsf.cdac.org.in:/storage3/G
arudaAdmin/2011-05-18/sample_file.11243.0"
------------------------------------------------------------
```

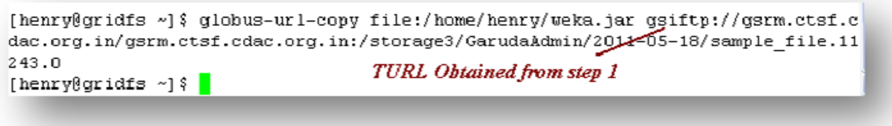
Fig. 6.13: clientSRM command to make a ptp request.

Step 2: Transfer the file

"***globus-url-copy***" command is used to transfer the file to GSRM server. This command requires a Transport URL (TURL), which is obtained from step 1 (Refer Fig. 6.13). The complete command is shown in Fig. 6.14.

Command:

globus-url-copy <source_path> <TURL>



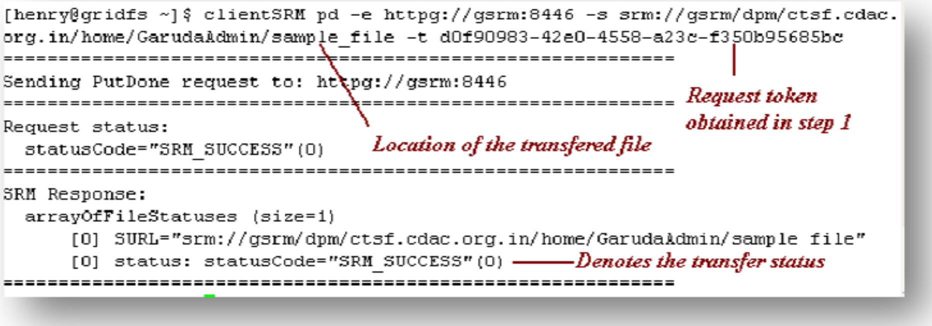
```
[henry@gridfs ~]$ globus-url-copy file:/home/henry/weka.jar gsiftp://gsrm.ctsf.c
dac.org.in/gsrm.ctsf.cdac.org.in:/storage3/GarudaAdmin/2011-05-18/sample_file.11
243.0
[henry@gridfs ~]$
```

TURL Obtained from step 1

Fig. 6.14: globus-url-copy is used to transfer the file to GSRM

Step 3: Verify the transfer

The transfer operation that performed in step 2 can be verified using the '***put done (pd)***' function of the ***clientSRM*** command. As shown in Fig. 6.15, this command requires a request token, which is obtained during step 1 (Refer Fig. 6.13).



```
[henry@gridfs ~]$ clientSRM pd -e httpg://gsrm:8446 -s srm://gsrm/dpm/ctsf.cdac.
org.in/home/GarudaAdmin/sample_file -t d0f90983-42e0-4558-a23c-f350b95685bc
============================================================
Sending PutDone request to: httpg://gsrm:8446
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
  arrayOfFileStatuses (size=1)
      [0] SURL="srm://gsrm/dpm/ctsf.cdac.org.in/home/GarudaAdmin/sample file"
      [0] status: statusCode="SRM_SUCCESS"(0)
============================================================
```

Request token obtained in step 1

Location of the transfered file

Denotes the transfer status

Fig. 6.15: Verifying the transfer performed in step 2.

How to download a file from GSRM using clientSRM command?

Downloading a file from GSRM server involves two steps,

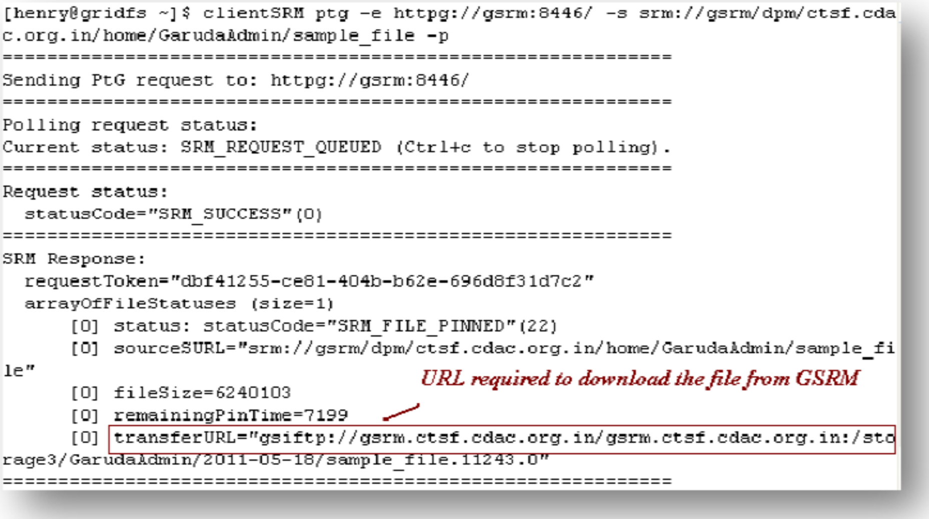
Step 1: Prepare to get request from GSRM (***clientSRM ptg***)
Step 2: Download the file. (***globus-url-copy***)

Step 1: Prepare to get request from GSRM

As shown in the Fig. 6.16, "put to get request (ptg)" function of the clientSRM command is used to make get request to GSRM server.

Command:

clientSRM ptg -e <GSRM_End_Point> -s <SURL_to_get_file> -p

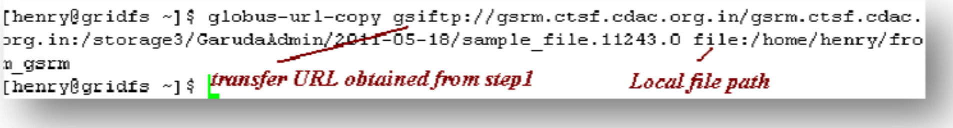


```
[henry@gridfs ~]$ clientSRM ptg -e httpg://gsrm:8446/ -s srm://gsrm/dpm/ctsf.cda
c.org.in/home/GarudaAdmin/sample_file -p
============================================================
Sending PtG request to: httpg://gsrm:8446/
============================================================
Polling request status:
Current status: SRM_REQUEST_QUEUED (Ctrl+c to stop polling).
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
  requestToken="dbf41255-ce81-404b-b62e-696d8f31d7c2"
  arrayOfFileStatuses (size=1)
      [0] status: statusCode="SRM_FILE_PINNED"(22)
      [0] sourceSURL="srm://gsrm/dpm/ctsf.cdac.org.in/home/GarudaAdmin/sample_fi
le"
      [0] fileSize=6240103
      [0] remainingPinTime=7199
      [0] transferURL="gsiftp://gsrm.ctsf.cdac.org.in/gsrm.ctsf.cdac.org.in:/sto
rage3/GarudaAdmin/2011-05-18/sample_file.11243.0"
============================================================
```

Fig. 6.16, Using ptg function to make a get request to GSRM

Step 2: Download the file from GSRM

As shown in Fig. 6.17, ***globus-url-copy*** command is used to get the file from GSRM. This command requires the ***transfer URL*** obtained in step 1 (Refer Fig. 6.16)

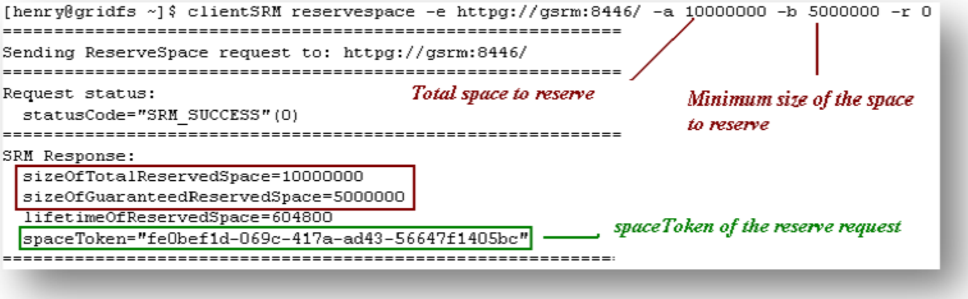


```
[henry@gridfs ~]$ globus-url-copy gsiftp://gsrm.ctsf.cdac.org.in/gsrm.ctsf.cdac.
org.in:/storage3/GarudaAdmin/2011-05-18/sample_file.11243.0 file:/home/henry/fro
m_gsrm
[henry@gridfs ~]$
```

Fig. 6.17: globus-url-copy command is used to download the file

How to reserve a space in GSRM?

As shown in Fig. 6.18, "***reservespace***" function of the ***clientSRM*** command is used to reserve space in GSRM. This function is used to reserve a space in advance for the upcoming requests to get some guarantee on the file management. Unit of the space should be specified in bytes. Fig. 6.18, reserves a total space of 10MB and requests for a minimum space of size 5MB. The returned space token have to be used in all the requests that work on this reserved space (green colored rectangle in the image).



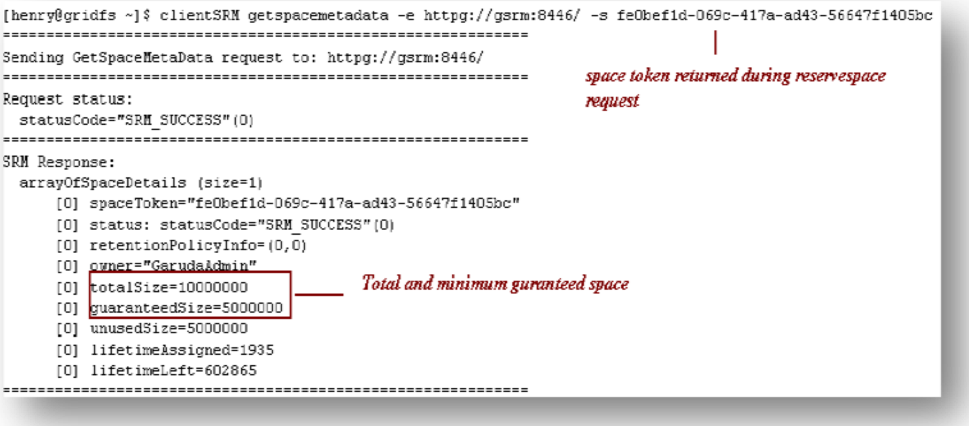
```
[henry@gridfs ~]$ clientSRM reservespace -e httpg://gsrm:8446/ -a 10000000 -b 5000000 -r 0
============================================================
Sending ReserveSpace request to: httpg://gsrm:8446/
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
  sizeOfTotalReservedSpace=10000000
  sizeOfGuaranteedReservedSpace=5000000
  lifetimeOfReservedSpace=604800
  spaceToken="fe0bef1d-069c-417a-ad43-56647f1405bc"
============================================================
```

Total space to reserve
Minimum size of the space to reserve
spaceToken of the reserve request

Fig. 6.18: Reserve space in GSRM

How to view the information of a reserved space?

As shown in Fig. 6.19, "***getspacemetadata***" function of the ***clientSRM*** command is used to view the information of a reserved space. This function requires a ***space token (spaceToken)*** that was returned during "***reservespace***" request (Refer green colored rectangle in Fig 6.18).



```
[henry@gridfs ~]$ clientSRM getspacemetadata -e httpg://gsrm:8446/ -s fe0bef1d-069c-417a-ad43-56647f1405bc
=================================================================
Sending GetSpaceMetaData request to: httpg://gsrm:8446/
=================================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
=================================================================
SRM Response:
  arrayOfSpaceDetails (size=1)
      [0] spaceToken="fe0bef1d-069c-417a-ad43-56647f1405bc"
      [0] status: statusCode="SRM_SUCCESS"(0)
      [0] retentionPolicyInfo=(0,0)
      [0] owner="GarudaAdmin"
      [0] totalSize=10000000
      [0] guaranteedSize=5000000
      [0] unusedSize=5000000
      [0] lifetimeAssigned=1935
      [0] lifetimeLeft=602865
=================================================================
```

space token returned during reservespace request

Total and minimum guranteed space

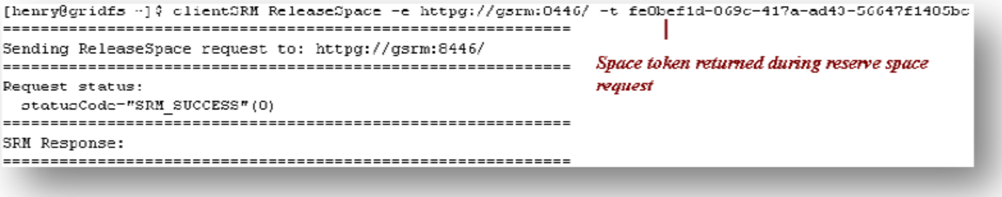
Fig. 6.19: View the information of a reserved space.

Note: The reserved space token can be made use during your file upload request to GSRM. This can be achieved using "***-t***" option to the "***prepare-to-put (ptp)***" function of the ***clientSRM*** command. Refer Fig. 6.13 for more information about "***ptp"*** function of a ***clientSRM*** command.

Note: Type "***clientSRM <function> -help***" to view detailed options for a particular function

How to release the reserved space in GSRM?

As shown in Fig. 6.20, "***ReleaseSpace***" function of a ***clientSRM*** is used to release the reserved space. This function requires "***spaceToken***" as an input, which was returned during the "reservespace" request (Refer the green colored rectangle in Fig. 6.18).



```
[henry@gridfs ~]$ clientSRM ReleaseSpace -e httpg://gsrm:8446/ -t fe0bef1d-069c-417a-ad43-56647f1405bc
============================================================
Sending ReleaseSpace request to: httpg://gsrm:8446/
============================================================
Request status:
  statusCode="SRM_SUCCESS"(0)
============================================================
SRM Response:
============================================================
```

Space token returned during reserve space request

Fig. 6.20: Release the reserved space

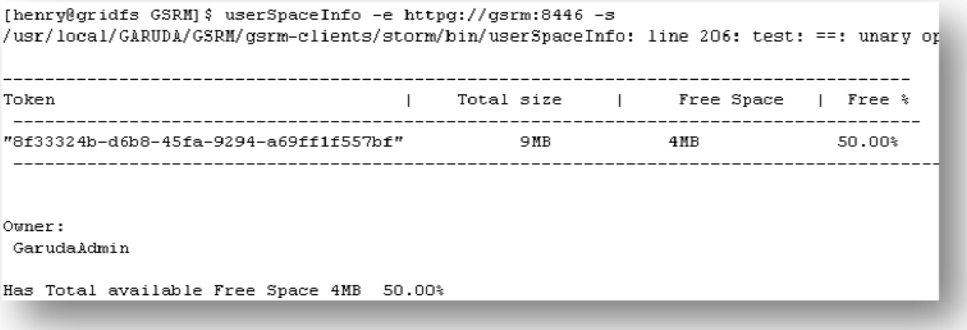
How to view the reserved space information?

As shown in Fig. 6.21, "***userSpaceInfo***" command can be used to view the reserved space information.

- It allows the user to view the respective space usage details in KB/MB/GB.
- User can get the reserved space details of all the space tokens associated with the user.
- User can also view overall space usage i.e. reserved and unreserved.

Command:

userSpaceInfo -e <GSRM_End_Point> -s



```
[henry@gridfs GSRM]$ userSpaceInfo -e httpg://gsrm:8446 -s
/usr/local/GARUDA/GSRM/gsrm-clients/storm/bin/userSpaceInfo: line 206: test: ==: unary op

--------------------------------------------------------------------------------------
Token                                  |    Total size     |     Free Space    |  Free %
 --------------------------------------------------------------------------------------
"8f33324b-d6b8-45fa-9294-a69ff1f557bf"           9MB           4MB              50.00%
 ----------------------------------------------------------------------------------------


Owner:
 GarudaAdmin

Has Total available Free Space 4MB  50.00%
```

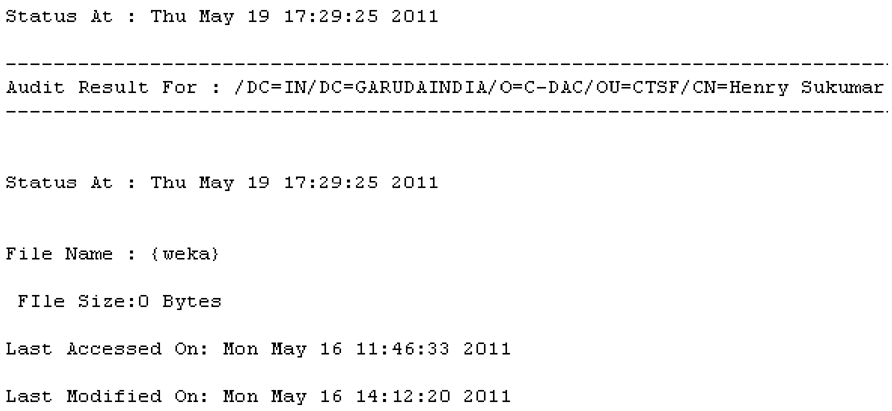
Fig 6.21: userSpaceInfo command to view reserved space tokens

How to view user audit information?

As shown in Fig. 6.22, "***selfAudit***" can be used to view audit information of the user data stored in GSRM. Audit information includes filename, file size, time last modified and last accessed.

Command:

selfAudit -s <hostname>



```
Status At : Thu May 19 17:29:25 2011

---------------------------------------------------------------------------
Audit Result For : /DC=IN/DC=GARUDAINDIA/O=C-DAC/OU=CTSF/CN=Henry Sukumar
---------------------------------------------------------------------------


Status At : Thu May 19 17:29:25 2011


File Name : (weka)

 FIle Size:0 Bytes

Last Accessed On: Mon May 16 11:46:33 2011

Last Modified On: Mon May 16 14:12:20 2011
```

Fig 6.22: selfAudit command to view the audit information

Does GSRM have a web interface?

Yes, GSRM web client can be accessed from

***https://gsrm.ctsf.cdac.org.in*.**

https://srmhyd.cdac.in

Note: The above URL can be accessed only via GARUDA NKN Network.

Where can I find complete user manual for GSRM?

GSRM User manual can be downloaded from,

http://portal.garudaindia.in/GARUDA-REPO/gsrmv1.1/gsrmv1.1_Manuals/gsrmv1.1_usermanual.pdf

Whom should I contact, if I need a support for GSRM?

You can reach our GSRM team at ***080-66116692*** and ***080-66116457***

GSRM End Points in GARUDA:

- ***Bangalore: httpg://gsrm.ctsf.cdac.org.in:8446***
- ***Hyderabad: httpg://srmhyd.cdac.in:8446***

Note: The protocol used for contacting the end point is "***httpg***"

Chapter 7

Grid Support

Chapter 7: Grid Support

How will I log my support request?

1. Go to ***http://gridsupport.garudaindia.in***
2. Login with your GARUDA Account.
3. Raise ticket in appropriate queue.

Which username and password should I use for accessing Grid Support web page?

You must have a GARUDA account for accessing Grid Support. Refer ***Fig. 1.8.***

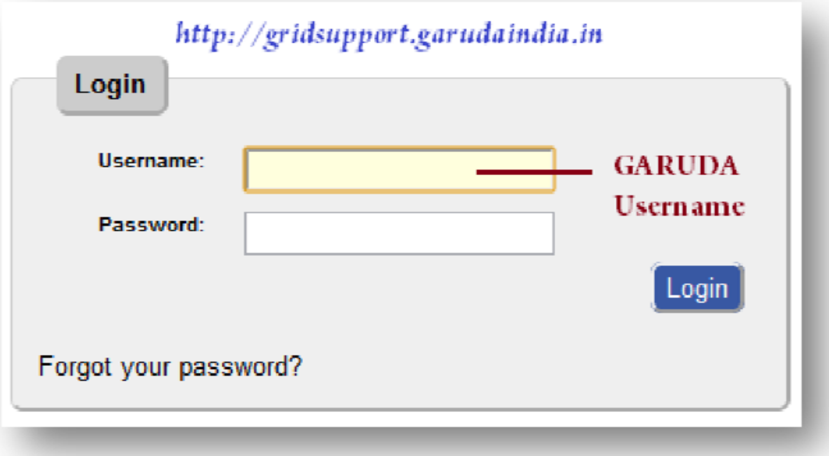


Fig. 7.1: GARDUDA RT (Request Tracker) Login page

How to raise a problem ticket?

1. Select the queue for which you would like to raise a ticket.
2. Click "***New ticket in***" button as shown in Fig. 7.2.

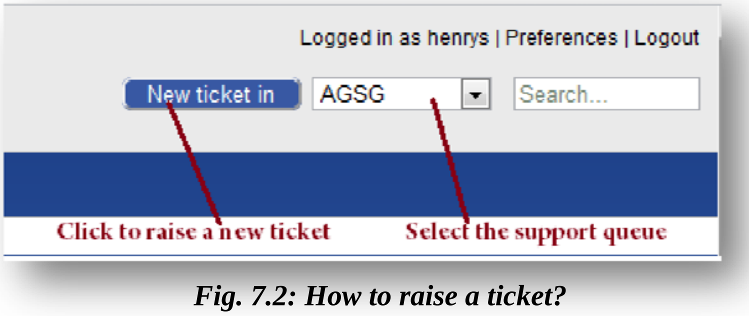
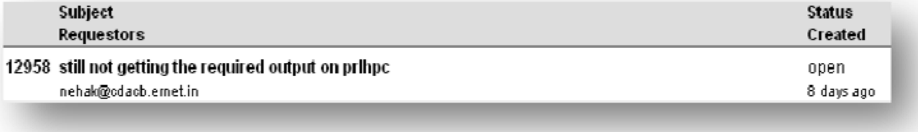

Logged in as henrys | Preferences | Logout
New ticket in
AGSG
Search...
Click to raise a new ticket
Select the support queue

Fig. 7.2: How to raise a ticket?

How to check the status of my ticket?

User can check the status of their ticket using the ticket ID. E.g. ***[Ticket Id #12958]*** as shown in Fig. 7.3.



	Subject Requestors	Status Created
12958	still not getting the required output on prlhpc nehak@cdacb.ernet.in	open 8 days ago

Fig. 7.3: Status of the ticket

What are the support queues available?

Sl.no	Project Name	Support queue name	Email
2.	Any Grid related problem	GDeployment	***grid-help@cdacb.ernet.in***
3.	Grid Portal	Portal	***rt-gp@cdacb.ernet.in***

4.	**Service Oriented Architecture**	**SOA**	**rt-soa@cdacb.ernet.in**
5.	GARUDA VOMS	Voms	**voms@cdacb.ernet.in**
6.	Grid IDE Tool	GridIDE	**rt-gide@cdacb.ernet.in**
7.	OSDD	OSDD	**rt-osdd@cdacb.ernet.in**
8.	OSDD Galaxy	Osddgalaxy	**rt-osddgalaxy@cdacb.ernet.in**
9.	GARUDA Network Related	Netops	**netops@cdacb.ernet.in**
10.	Paryavekshanam	Gridmon	**rt-gmon@cdacb.ernet.in**
11.	Gridhra	Gridhra	
12.	Automatic Grid Service Generator	AGSG	**rt-agsg@cdacb.ernet.in**
	GARUDA SRM	GDS	**rt-gds@cdacb.ernet.in**
13.	Not sure about Queue	Other	**rt@cdacb.ernet.in**

Table 7.1: GARUDA support queues

Can I send my support request via email?

Yes, refer Table 7.1, for corresponding email addresses.

Can I speak to the GARUDA Operation team over phone?

Why not! We will be glad to speak to you. The support phone nos. are given below.

Grid Support: 080-66116511 | 080-66116472 | 080-66116474

Network Support: 080-66116510 | 080-66116473

Grid Portal: 080-66116493 | 080-66116494 | 080-66116459

GSRM: 080-66116692 | 080-66116457

Help us to serve you better:

We request you to kindly spare some of your valuable time and fill in the online survey form. Your input would help us to improve our services and allow us to serve you better.

Online survey form: [***http://gridsupport.garudaindia.in/survey/index.php?sid=31228***](http://gridsupport.garudaindia.in/survey/index.php?sid=31228)

Whom should I contact, if I have any question regarding this book?

You can send a mail to ***rt@cdacb.ernet.in.***

Tips for Quick Response:

- *Raise tickets in appropriate queue.*
- *Set your priority level to indicate the urgency of the problem.*
- *Describe your problem clearly.*
- *Attach screen shots if applicable.*
- *Maintain the same subject name, until the ticket is resolved.*

Appendix I

Appendix I: Glossary

The following glossary defines some commonly used terms and acronyms associated with the GARUDA Environment.

A

AGSG – Automatic Grid Service Generator.

Authentication - A system says how you identify yourself to the computer. The goal behind an authentication system is to verify that the user is actually who they say they are.

Authorization - Authorization checks that authentic user have proper permission or not to access particular service.

C

Certificate - A certificate is a digital document that certifies that a certain public key is owned by a particular user. This document is signed by Certificate Authority (CA).

Certificate Authority - Certification authority (CA) is an entity that issues digital certificates for use by other parties.

CP/CPS – Certificate Policy and Privacy Statement.

CRIN PIN (Certificate Revocation and Identification Number) - The CRIN mail contains a pin code that the user must enter when revoking their own certificates. The user should be able to decrypt the message as they would have created the private key during their certificate request process. The message is encrypted using the public key in the certificate request.

D

DN – Distinguished Name. Unique name for each user in the grid.

E

End Entity Certificate – User certificate or Host certificate used by the end-user is called as End-Entity Certificate.

G

GARUDA – Global Access to Resources Using Distributed Architecture.

GARUDA Account – An account used for accessing GARUDA Grid.

GAP – Grid Access Portal

Grid Computing – Grid computing is a term referring to the combination of computer resources from multiple administrative domains to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. What distinguishes grid computing from conventional high performance computing systems such as cluster computing is that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed. Although a grid can be dedicated to a specialized application, it is more common that a single grid will be used for a variety of different purposes. Grids are often constructed with the aid of general-purpose grid software libraries known as middleware.

Globus Toolkit - The Globus Toolkit is a fundamental enabling technology for building grids that allow distributed computing power, storage resources, scientific instruments, and other tools to be shared securely across corporate, institutional, and geographic boundaries.

GRIDFS – Name of the machine that acts a GARUDA HEAD NODE (gridfs.ctsf.cdac.org.in).

GARUDA INTERFACE NODE – NODE located at each partner institution which connects to garuda.

Grid Portal - Grid portal is for submitting sequential, parallel. This portal provides a web interface for viewing the resources and submitting and monitoring jobs.

GRAM – Grid Resource and Allocation Manager - GRAM is a Globus project that produces technologies which enable users to locate, submit, monitor and cancel remote jobs on Grid-based compute resources. GRAM enables remote execution management in contexts for which reliable operation, stateful monitoring, credential management, and file staging are important.

GridFTP - GridFTP is a Globus project that produces high-performance, secure, reliable data transfer technologies optimized for high-bandwidth wide-area networks.

GSISSH - command to securely login to a remote machine.

GSISCP - command to securely copy data to a remote machine.

GSRM – GARUDA Storage Resource Manager.

GRIDWAY - The GridWay Metascheduler enables large-scale, reliable and efficient sharing of computing resources: clusters, supercomputers, stand-alone servers… It supports different LRM systems (PBS, SGE, LSF, and Condor) within a single organization or scattered across several administrative domains.

H

Host Certificate – Certificate issued by Certificate Authority for a particular host to participate in Grid.

I

IGCA - The Indian Grid Certification Authority (IGCA) provides X.509 certificates to support the secure environment in grid computing. IGCA is an accredited member of the APGridPMA (Asia Pacific Grid Policy Management Authority) for Grid Authentication.

M

Middleware – Middleware is an interface between the user and resources. Globus Toolkit is one of the grid middleware.

MyProxy Credential Repository - MyProxy is a Globus project that develops software for managing X.509 Public Key Infrastructure security credentials (certificates and private keys).

MPI – Message Passing Interface (**MPI**) is an API specification that allows processes to communicate with one another by sending and receiving messages. It is typically used for parallel programs running on computer clusters and supercomputers, where the cost of accessing non-local memory is high.

N

NKN – National Knowledge Network. The backbone of GARUDA Grid.

P

Public Key – Certificate key to be shared with other party for communication.

Private Key – Certificate key that should not be shared with anyone.

PKI - Enables users to securely and privately exchange data and money through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority.

.PEM (Privacy Enhanced Mail) – Certificate format used in grid.

.P12 & PFX - PKCS#12, contains certificate(s) (public) and private keys (password protected)

PKCS – Public Key Cryptographic System

Proxy Certificate – A short lived certificate for accessing grid services. Generated using public and private keys.

Paryaveekshanam – Grid monitoring application for GARUDA.

Q

QOS - Quality of Service

R

Revoked Certificate - A certificate is said to be invalid once it is revoked.

Re-key Certificate - The lifetime of IGCA user certificates and host certificates are valid for 12 months. User must rekey the related certificate when his or her certificate will expire. After you rekey the certificate, you will own the brand-new certificate and private key. The previous certificate and private key will be revoked.

Registration Authority - A Registration Authority (RA) is an authority in a network that verifies user request for a digital certificate and tells the Certificate Authority (CA) to issue certificate.

S

SOA – Service Oriented Architecture.

U

User Certificate – Certificate issued to user for accessing Grid.

V

VOMS - VOMS is a system for managing authorization data within multi-institutional collaborations. VOMS provides a database of user roles and capabilities and a set of tools for accessing and manipulating the database and using the database contents to generate Grid credentials for users when needed.

W

WS-GRAM - The Grid Resource Allocation and Management (GRAM) web service provides a single interface for requesting and using remote system resources for the execution of "jobs". The most common use of GRAM is remote job submission and control. It is designed to provide a uniform, flexible interface to job scheduling systems.

Bibliography

References:

1. Garuda Official Website : http://garudaindia.in
2. Grid Portal : http://portal.garudaindia.in
3. Indian Grid Certification Authority : http://ca.garudaindia.in
4. GARUDA VOMS Registration : http://voms.garudaindia.in
5. Grid Support : http://gridsupport.garudaindia.in
6. Globus Toolkit : http://globus.org
7. Gridway : http://gridway.org
8. GARUDA GridFTP Client : http://portal.garudaindia.in/gridftp-gui
9. GSRM : http://storm.forge.cnaf.infn.it/documentation/client_examples
10. GSRM : http://csngwinfo.in2p3.fr/mediawiki/index.php/SRM_StoRM
11. GSRM : http://portal.garudaindia.in/GARUDA-REPO/gsrmv1.1/gsrmv1.1_Manuals/gsrmv1.1_usermanual.pdf

Important Links and Email-ids to remember

Garuda Website: http://garudaindia.in
grid-help@cdacb.ernet.in

IGCA: http://ca.garudaindia.in
igca@cdacb.ernet.in

Grid Support:
http://gridsupport.garudaindia.in